A Multi-objective Genetic Algorithm for Peptide Optimization

Dissertation

zur Erlangung des Doktorgrades der Naturwissenschaften
(Dr. rer. nat.)

dem Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg
vorgelegt von

Susanne Rosenthal
geboren in Köln

Marburg, im Mai 2016

Vom Fachbereich Mathematik und Informatik der
Philipps-Universität Marburg als Dissertation am

_____

angenommen.

Erstgutachter: Prof. Dr. Bernd Freisleben, Philipps-Universität Marburg

Zweitgutachter: Prof. Dr. Thomas Bäck, Universität Leiden, Niederlande

Tag der mündlichen Prüfung am 21.11.2016.

Abstract

The peptide-based drug design process requires the identification of a wide range of candidate molecules with specific biological, chemical and physical properties. The laboratory analysis in terms of *in vitro* methods for the discovery of several physiochemical properties of theoretical candidate molecules is time- and cost-intensive. Hence, *in silico* methods are required for this purpose. Metaheuristics like evolutionary algorithms are considered to be adequate *in silico* methods providing good approximate solutions to the underlying multi-objective optimization problems. The general issue in this area is the design of a multi-objective evolutionary algorithm to achieve a maximum number of high-quality candidate peptides that differ in their genetic material, in a minimum number of generations.

A multi-objective evolutionary algorithm as an *in silico* method of discovering a large number of high-quality peptides within a low number of generations for a broad class of molecular optimization problems of different dimensions is challenging, and the development of such a promising multi-objective evolutionary algorithm based on theoretical considerations is the major contribution of this thesis. The design of this algorithm is based on a qualitative landscape analysis applied on a three- and four-dimensional biochemical optimization problem. These problems are generic in the sense that sequence-derived structural and physiochemical features of peptides are calculated from amino acid descriptor values. This is a widely used method in the area of machine learning to predict peptide features and molecular interactions.

Qualitative and quantitative landscape analysis are common techniques to gain insights into the landscape structure and the potential difficulties in guiding the search process of an evolutionary algorithm in the direction of the optimal solutions. Landscape analysis methods characterizing real-valued multi-objective landscapes are challenging. The transfer and the re-definition of the established landscape properties which arise in single-objective landscape analysis and are applied to molecular multi-objective landscapes is another contribution of this thesis. The conclusions drawn from the empirical landscape analysis of the three- and four-dimensional optimization problem result in the formulation of

hypotheses regarding the types of evolutionary algorithm components which lead to an optimized search performance for the purpose of peptide optimization.

Starting from the established types of variation operators and selection strategies, different variation operators and selection strategies are proposed and empirically verified on the three- and four-dimensional molecular optimization problem with regard to an optimized interaction and the identification of potential interdependences as well as a fine-tuning of the parameters. Moreover, traditional issues in the field of evolutionary algorithms such as selection pressure and the influence of multi-parent recombination are investigated. The experiments are evaluated according to the evolutionary algorithm objectives: convergence, diversity and relative quality of the non-dominated solutions. Another contribution of this thesis is the presentation of a convergence indicator which is statistically reasonable, does not require the knowledge of a Pareto optimal solution set and allows the comparison of differently sized solution sets. The properties of this convergence indicator are discussed. Based on this indicator, a further measurement reflecting the relative quality of the non-dominated solutions is proposed. A scaled version of this convergence indicator is further used and investigated as a selection criterion in a selection strategy.

Zusammenfassung

Die Identifikation einer großen Bandbreite an potentiellen Molekülen mit spezifischen biologischen, chemischen und physikalischen Eigenschaften ist ein wesentlicher Bestandteil der Peptide-basierten Wirkstoffentwicklung. Die *in vitro* Analyse potentieller Wirkstoffkandidaten ist zeit- und kostenintensiv. Daher sind für diesen Zweck *in silico* Methoden erforderlich. Metaheuristiken wie evolutionäre Algorithmen sind als hinreichend gute *in silico* Verfahren zur approximativen Lösung der zugrundeliegenden multiobjektiven Optimierungsprobleme bekannt. Die generelle Problemstellung in diesem Bereich ist die Entwicklung eines multiobjektiven evolutionären Algorithmus mit dem Ziel, eine möglichst hohe Anzahl an potentiellen hoch-qualifizierten Peptiden, die sich in ihrem genetischen Material deutlich unterscheiden, in einer möglichst geringen Anzahl an Generationen zu erhalten.

Ein multiobjektiver evolutionärer Algorithmus als *in silico* Methode, der eine große Anzahl an hoch qualifizierten Peptiden in einer geringen Anzahl an Generationen für eine breite Klasse von molekularen Optimierungsproblemen verschiedener Dimensionen detektiert, ist herausfordernd und die Entwicklung eines solchen vielversprechenden multiobjektiven Algorithmus unter theoretischen Betrachtungen ist der Hauptbeitrag dieser Arbeit. Die Konstruktion dieses Algorithmus basiert auf einer qualitativen Analyse der Fitnesslandschaft angewandt auf ein drei- und vierdimensionalen Optimierungsproblems. Diese Optimierungsprobleme sind generisch in der Hinsicht, dass Sequenz-abgeleitete strukturelle und physikalisch-chemische Peptideigenschaften anhand von Deskriptorenwerten der Aminosäuren berechnet werden. Dies ist eine weit verbreitete Methode im Bereich des maschinellen Lernens um Peptideigenschaften und molekulare Interaktionen vorherzusagen.

Die qualitative und quantitative Analyse der Fitnesslandschaft sind übliche Techniken zur Analyse der Fitness-Landschaft-Struktur und der potentiellen Schwierigkeiten bei der Lenkung des Suchprozesses eines evolutionären Algorithmus in Richtung der optimalen Lösungen. Analysemethoden für Fitness-Landschaften, welche reell-wertige multiobjektive Fitness-Landschaften analysieren, sind eine Herausforderung. Der Transfer und die Neudefinition der eta-

blierten Fitness-Landschaft-Eigenschaften, welche aus der singulär-objektiven Fitness-Landschaft-Analyse hervorgehen und auf die drei- und vierdimensionale Landschaft übertragen werden, ist ein weiterer Beitrag dieser Arbeit. Die Schlussfolgerungen aus der empirischen Fitness-Landschaft-Analyse des drei- und vierdimensionalen Optimierungsproblems resultieren in der Formulierung von Hypothesen hinsichtlich der Arten von evolutionären Algorithmenkomponenten, welche zur optimierten Suchperformanz des evolutionären Algorithmus zum Zweck der Peptidoptimierung führen. Ausgehend von den etablierten Arten der Variationsoperatoren und Selektionsstrategien werden verschiedene Variationsoperatoren und Selektionsstrategien vorgestellt und empirisch anhand des drei- und vierdimensionalen Optimierungsproblems im Hinblick auf eine optimale Interaktion und gegenseitige Abhängigkeit sowie einer Feineinstellung der Parameter getestet. Darüber hinaus werden die traditionellen Fragestellungen im Bereich der evolutionären Algorithmen wie Selektionsdruck und der Einfluss von mehreren Eltern bei der Rekombination untersucht. Die Experimente werden hinsichtlich Zielsetzungen des evolutionären Algorithmus ausgewertet: Konvergenz, Diversität und relative Qualität der nicht-dominierten Lösungen. Ein weiterer Beitrag dieser Arbeit ist die Vorstellung eines solchen Konvergenzindikators, der statistisch sinnvoll, keine Referenzmenge an Pareto optimalen Lösungen benötigt und den Vergleich von unterschiedlich großen Lösungsmengen ermöglicht. Die Eigenschaften dieses Konvergenzindikators werden diskutiert. Basierend auf diesem Indikator wird ein weiteres Maß vorgestellt, welches die relative Qualität der nicht-dominierten Lösungen reflektiert. Eine skalierte Version dieses Konvergenzindikators wird darüber hinaus als Selektionkriterium in einer Selektionsstrategie angewendet und untersucht.

Erklärung

Ich versichere, dass ich meine Dissertation

A Multi-objective Genetic Algorithm for Peptide Optimization

selbständig, ohne unerlaubte Hilfe angefertigt und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe. Die Dissertation wurde in der jetzigen oder einer ähnlichen Form noch bei keiner anderen Hochschule eingereicht und hat noch keinen sonstigen Prüfungszwecken gedient.

Susanne Rosenthal
Marburg, im Mai 2016

Acknowledgment

The research work presented in this thesis comprises a working period of five years. This was a time of great gladness as well as personal challenge and progress. However, this work would not be possible without the support and supervision of several people whom I would like to thank at this point:

Firstly, I would like to thank Prof. Dr. Bernd Freisleben for his willingness to support my dissertation project and his helpful comments in the cooperative publications.

Secondly, I would like the thank Prof. Dr. Thomas Bäck for his willingness to be the co-examiner of this thesis and his assistant professor Dr. Michael Emmerich for his inspirational conversations.

A special thank goes to my supervisor PD Dr. Markus Borschbach for his confidence to entrust me with this dissertation project and for introducing me into this research field with his infectious enthusiasm. Furthermore, I would like to thank him for his continuous and warm supervision regarding this thesis as well as in my position as a research assistant, which he offered to me. This is the reason why this time became the period of greatest personal development.

A further thanks go to my colleagues. First of all, I would like to thank Christian Linder for his programming support and the fruitful discussions. Furthermore, I would like to thank Nail El-Sourani and Sascha Hauke for their introduction into the OPTOPROBE project.

Last but not least, a sincerely thank goes to my family for their moral support during this challenging time of my promotion period.

This PhD is partly supported by the German Ministry of Education and Research (BMBF) by founding the project OPTOPROBE.

Table of Contents

List of Acronyms

1 Introduction

1.1 Motivation

Drug development is a systematical and multi-level step that has its starting point in pre-clinical research. Drugs are developed with the aim of treating a disease safely and effectively. The drug discovery process consists of four main steps: target identification, target validation, lead identification and lead optimization. The target identification step analyzes the target related to the disease to understand how the target influences a health body instead of a diseased one. The target validation steps analyzes if the target is directly involved in the disease process. The lead identification step identifies and evaluates molecules regarding the biological activity on the disease target or function. In the fourth step, the lead is optimized as a potential drug candidate. The most challenging step of the drug design process is the identification of a target peptide or protein. These identified peptides usually miss the high specificity that is essential to avoid the mentioned negative side effects like toxicity, digestion or expulsion. Additionally, they miss the suitable molecular features that are important for successful drugs. Beneath the traditional biochemical experiments, the prediction of the molecular properties of peptides and proteins is one of the main application fields for chemoinformatics [109].
In general, peptides play an important role in the area of biological, medical and pharmaceutical research. Due to their physiological features and biochemical activity, the application fields of peptides are therapeutic and diagnostic interventions. The favorable characteristics of peptides include the generally high activity on their target receptor, a good target specificity and a relatively small mass. This results in high effectiveness at a low dose. Furthermore, peptides tend to have low adverse side effects and have a high binding affinity on the target receptor. Peptides are physiologically accepted as natural biological products that are associated with less accumulation in body issue and fewer toxic adverse effects [121].
The focus in the area of drug design is the identification of peptides that opti-

mize several molecular properties and reduce the following - for the purpose of drug design challenging - pharmaceutical features of native peptides: Peptides typically have a very short half-life, this means that they have a general active time of 2 to 30 minutes before they break down to reuse the amino acids for building blocks. Furthermore, peptides are metabolically unstable due to the protease cleavage of the backbones. Peptides usually cannot be administered orally as this leads to destruction of the peptides by the digestive system to ineffective amino acids. Another potentially unfavorable property is the solubility of peptides in aqueous solutions, meaning that peptides are only stable in aqueous solutions for a few days and this limits their utility. These challenging pharmaceutical properties of peptides are well known [51], [27].
Designer peptide drugs elude the negative pharmacological properties of native peptides and impress by optimized biochemical properties. Especially small peptides with a length of 10 to 20 amino acids are interesting since they combine the molecular requirements of reduced ligand size while maintaining specificity in biological interactions [130]. Drug development requires the characterization of peptides. Since the synthesis of peptides and the laboratory analysis are cost-intensive, *in silico* methods are required to design short peptide sequences with diverse optimized molecular features. In this field, multi-objective evolutionary algorithms are effective methods [158], [116], [94], [134]. Singh reports a specific property of a Genetic Algorithm (GA) in the area of molecular optimization [145]: a genetic algorithm is introduced that optimizes active leads. From the 20^6 possible hexapeptides, only less than 300 had to be synthesized, found by the GA within only five generations as optimized leads. More precisely, the experiments show that each successive generation provides an optimization progress by attaining high quality leads in each of the successive generation.
This thesis is generally motivated by this work of Singh. Design considerations are presented regarding the composition of a Multi-Objective Evolutionary Algorithm (MOEA) with the final aim of multi-objective molecular optimization under specific conditions: Highly diverse peptide sequences with multiple optimized molecular features have to be recovered within a low number of generations. The recovering of optimized peptides in each succeeding generation (overall less than 20 generations) is intended and termed early convergence in this thesis. Less work has been done so far to empirically analyze MOEAs regarding the effect of early convergence while maintaining a genetic diversity within the candidate solutions, in particular in the area of peptide optimization.

1.2 Terminology

The main goal of an Evolutionary Algorithm (EA) is to find optimal solutions for the underlying optimization problems by imitating the processes of Darwinian evolution. The EA terminology refers to the Darwinian evolution terminology and is transferred to the context of search heuristic as follows:

- **Chromosome:** Chromosomes are encoded strings of parameters (binary, float, character, etc.) and represent possible solutions of the optimization problem. Chromosomes are also termed as genotypes.
- **Gene:** A gene is a position or a set of positions in a chromosome.
- **Allele:** Possible values of a gene from a fixed set of symbols are termed alleles.
- **Individuals:** Individuals are candidate solutions to an optimization problem.
- **Population:** A set of structural similar individuals is termed population.
- **Fitness:** The quality of an individual is measured by a function. The function value is an indicator for the quality of a solution and is termed fitness.
- **Recombination:** The recombination is an operator that is used on two or more individuals (in this context termed parents) to generate new individuals (termed as offspring).
- **Mutation:** The mutation is an operator that is used on each individual to modify it randomly.
- **Selection:** The selection decides which individual is becoming a parent for recombination or which individual finds its way in the succeeding population.
- **Generation:** An iteration step in EAs consists of recombination, mutation and selection with the aim of producing a new population, which is termed generation.

Generally, EAs differ from each other with respect to their individual representation and in the composition of the used operators for mutation, recombination and selection.

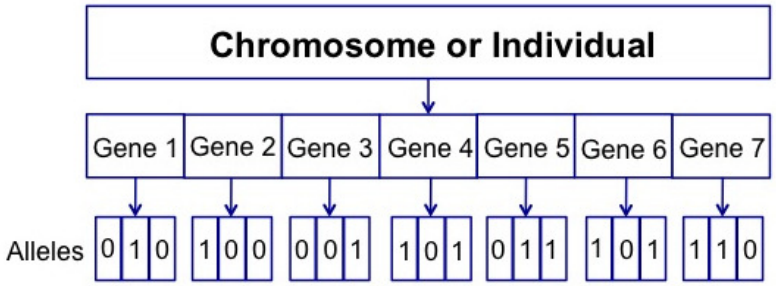

Fig. 1.1: Exemplary illustration of a chromose or individual.

In the case of a Multi-objective Optimization Problem (MOP), a number k of objective functions have to be optimized simultaneously. Without loss of generality, it is assumed that all objectives have to be minimized. A maximization problem is easily converted into a minimization problem by multiplying the objective functions with minus one. The minimization of a MOP with k objective functions is defines as:

$$\min_{x \in Q} \{F(x)\} \tag{1.1}$$

where Q is the decision (variable) space and $F(x)$ is defined as the objective vector consisting of k objective functions $F : Q \longrightarrow \mathbb{R}^k$, $F(x) = (f_1(x), ..., f_k(x))$ with $f_i : Q \longrightarrow \mathbb{R}$. $F(x)$ is denoted as the objective space.
In real-life application problems, objectives are conflicting with each other. It is nearly impossible to find a solution that optimizes all objectives perfectly. Therefore, a solution is termed optimal if it satisfies all objective functions sufficiently well and there is no other solution that dominates it. For a unique understanding of a Multi-Objective Optimization (MOO) concept, some definitions have to be clarified:

Definition 1 ***Pareto-dominance***
A vector $u \in \mathbb{R}^k$ is said to dominate $v \in \mathbb{R}^k$ (denoted as $u \prec v$) in the case of the MOP (1.1) if and only if u is partially less than v:
$\forall i \in \{1, ..., k\}$*:* $u_i \leq v_i$ *and there exists at least one* $i \in \{1, ..., k\}$*:* $u_i < v_i$.
A vector u is said to weakly dominate v (denoted as $u \preceq v$) if
$\forall i \in \{1, ..., k\}$*:* $u_i \leq v_i$.

Definition 2 ***Pareto-optimal***
Consider a set of decision solutions $X \in Q$ of the MOP (1.1). The set X is termed a global Pareto-optimal set if $\forall x \in X, \nexists y \in Q : F(y) \prec F(x)$.

In contrast to the Pareto dominance, two vectors are termed indifferent if one of them is superior in some dimensions, but worse in other dimensions - with regard to the optimization problem. Especially two Pareto-optimal points are indifferent to each other.

Definition 3 ***Indifference***
A vector $u \in \mathbb{R}^k$ is indifferent with regard to a vector $v \in \mathbb{R}^k$ (denoted as $u \sim v$) precisely when neither u dominates v nor v dominates u.

The main goal of a MOEA is to find the Pareto optimal set. The Pareto optimal set refers to the variable space, whereas the Pareto-optimal front (briefly Pareto front) refers to the set of objective vectors corresponding to the solutions in the Pareto set.

Definition 4 ***Pareto-optimal set***
For a given MOP ((1.1)), the Pareto-optimal set (P^*) is defined as:
$P^* := \{x \in Q | \nexists x' \in Q : F(x') \prec F(x)\}$

Definition 5 ***Pareto front***
For a given MOP ((1.1)) and the Pareto-optimal set (P^*) the Pareto-optimal front (PF^*) is defined as: $PF^* := \{F(x) | x \in P^*\}$

EAs as MOO approaches are evolved to fulfill the following three general and conflicting goals that have first been declared by Zitzler et al. [181]:

1. The approximate Pareto-optimal solution set has to be as close as possible to the Pareto front. In the best case, the approximate Pareto-optimal solution set is a subset of the Pareto front.

2. Individuals in the approximate Pareto-optimal solution set have to be uniformly distributed and widespread over the Pareto front.

3. The approximate Pareto-optimal solution set has to capture the whole spectrum of the Pareto front. This requires investigating solutions at the extreme ends of the objective function space.

The first goal is referred to as convergence and the second goal is denoted as diversity throughout this thesis.

1.3 Research Issues

The focus of this thesis is the design of a MOEA for peptide optimization providing a wide range of high quality peptides within a very low number of generations. More precisely, a customized NSGA-II is designed with the purpose of optimized algorithm performance in terms of convergence, diversity and non-dominated solution quality. Since the configuration and types of the algorithm components have a large influence on this search process, a suitable choice of the components regarding the purpose of early convergence in the application field of peptide optimization is challenging. In the following, questions and hypotheses are raised that will be answered, confirmed or ruled out in this thesis based on theoretical considerations and empirical investigations. According to previous work [117], the variation operators mutation and recombination in a GA cannot be optimized independently regarding the mutation and recombination rates. Moreover, the optimal mutation rates depend on the existence of recombination in a GA procedure. Therefore, the following hypothesis and question are important:

(H1) The configuration and types of recombination and mutation operators as well as their interaction influence the search behavior of the proposed MOEA.

(Q1) Is it feasible to define a category of recombination operators or mutation operators - or at least a combination of these two operators - that result in efficient and robust MOEA performance for biochemical optimization problems?

A common topic in the EA research field is the improvement of algorithm performance by a variation of the parent number for recombination. The following common hypothesis has to be examined:

(H2) The increase of the parent number for recombination results in an improvement of the algorithm performance.

In addition to the basic variation operators, the selection procedure is also responsible for a suitable balance between exploration and exploitation within the search process. This leads to the following hypothesis.

(H3) A suitable selection procedure is configured under the aspects of chance and an appropriate balance between diversity preserving method and suitable assignment of selection probabilities to high quality solutions in

the way that these solution have a higher chance to be selected into the succeeding generation than lower qualified solutions.

The balance between the diversity preserving method and selection probability assignment to high quality peptides as well as the influence level of change have to be controlled by selection parameters. This leads to the next question:

(Q2) Is it possible to define a range of settings for the selection parameters which allow a further improvement of the MOEA performance?

Early convergence as the most important property of the proposed MOEA is defeated since an increase of the population size results in a steady improvement of the algorithm performance. Less work has been done so far to investigate the influence of the MOEA parameter population size as well as the interdependence between the population size and the selection procedure on the MOEA performance. Therefore, the following questions are further considered in this thesis:

(Q3) Do large populations steadily speed up the algorithm performance of the proposed MOEA for the biochemical MOPs?

(Q4) Is there a range of population sizes that is able to provide optimized algorithm performance?

(Q5) Is there a predictable impact between population size and selection strategy?

(Q6) More precisely, is it possible to justify a configuration rule for the selection parameters and the population size that provides an optimized MOEA performance for the biochemical optimization problems?

According to the three goals declared in the last subparagraph, the experiments of the configurations are evaluated with regard to convergence, diversity and relative non-dominated solution quality. A convergence indicator is developed and introduced for the evaluation and comparison of configurations with differently sized populations in a statistically reasonable way. Furthermore, an intuitive diversity is proposed that calculates the spread of the solutions within a population in a statistically reasonable way as well. Moreover, a measurement is proposed that evaluates relative non-dominated solution quality and is based on the convergence indicator.
The theoretical basis of the MOEA design is the consideration of the molecular MOP according to the following question:

(Q7) How difficult and complex are the multi-objective molecular optimization problems, why is it difficult and which are the characteristic features of these landscapes?

An answer to this question requires an analysis concept that analyzes the corresponding molecular landscapes according to their fundamental properties. The specific landscape properties are discussed with respect to the explorative and exploitative search properties of the proposed customized NSGA-II to improve the performance.

1.4 Contributions

This thesis comprises methodological contributions and contributions to the current state of knowledge within evolutionary algorithm arising out of the research tasks of the BMBF project OPTOPROBE [69]. Another starting point of research work in this thesis is the statement of Singh [145]: Each generation of an evolutionary process for lead optimization represents an optimization progress by discovering high qualified leads in each generation. The central research tasks of the project OPTOPROBE was the rational and efficient identification of suitable ligands as tumor markers and special fluorescent probes to label intracellular targets. The theoretical foundation of the evolutionary process as well as the empirical comparison and discussion of the concepts is the main contribution of this thesis. Based on the project objectives to design a multi-objective evolutionary algorithm for the identification of peptide-based candidate probes, this thesis undertakes fundamental research work in designing a suitable evolutionary concept founded on theoretical considerations. This thesis does not disclose any data or technical details of the OPTOPROBE project.

A single-objective evolutionary process has been developed to identify peptide ligands with specific characteristics to target glycostructures [130]. An important insight from this research work is the exponential fitness improvement within the first five generations of the evolutionary process. The fitness improvement is slowed down for the generations 5 to 10. This characteristic is termed early convergence in this thesis. The identification of suitable peptide ligands requires the fulfillment of multiple objectives. The specification of a multi-objective evolutionary process based on the single-objective model with similar characteristics is required. Thus, the main contribution of this thesis is

- a concept of a multi-objective evolutionary process for peptide optimization providing early convergence.

This concept is based on the results and the insights of a molecular landscape analysis performed on two generic and different dimensional molecular optimization problems. The results are generalized to improve the search performance of the evolutionary process. Thus, the methodological contribution of this thesis is

- a concept for a qualitative real-valued multi-objective molecular landscape analysis.

The influence of the component parameters of the evolutionary process on the optimization performance is empirically analyzed on two generic and different dimensional molecular optimization problems. The performance results are statistically evaluated regarding the convergence and diversity behavior. For the purpose of convergence analysis in a statistically reasonable way, the following methodological contribution is part of this thesis:

- introduction of a convergence indicator and discussion of its properties.

1.5 Publications

Different parts of the research presented in this thesis have been published in the proceedings of several conferences, a journal and a book chapter after a double-blind peer-reviewed process. The publications included in this thesis are listed below.

Paper 1
S. Rosenthal, N. El-Sourani, and M. Borschbach, „Introduction of a Mutation Specific Fast Non-dominated Sorting GA Evolved for Biochemical Optimization“, *Proceedings of the 9th International Conference on Simulated Evolution and Learning* (SEAL 2012), LNCS 7673, pp. 158-167, 2012

Paper 2
S. Rosenthal, N. El-Sourani, M. Borschbach, „Impact of Different Recombination Methods in a Mutation-Specific MOEA for a Biochemical Application“. L. Vanneschi, W.S. Bush, and M. Giacobini (eds.): *Proceedings of the 11th Eu-*

ropean Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO 2013) LNCS 7833, pp. 188-199, 2013

Paper 3
S. Rosenthal and M. Borschbach, „A Benchmark on the Interaction of Basic Variation Operators in Multi-Objective Peptide Design evaluated by a Three Dimensional Diversity Metric and a Minimized Hypervolume“, M. Emmerich et al. (eds.): *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*, pp. 139-153, 2013

Paper 4
S. Rosenthal, and M. Borschbach, „Fine-Tuning of Genetic Algorithm Aggregate Selection for Multi-Objective Biochemical Optimization“. Emmerich et al. (Eds.): *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*, pp. 41-46, 2013

Paper 5
S. Rosenthal and M. Borschbach, „Impact of Population Size and Selection within a Customized NSGA-II for Biochemical Optimization Accessed on the basis of the Average Cuboid Volume Indicator“, *Proceedings of the 6th International Conference on Bioinformatics, Computational Systems and Biotechnologies* (BIOTECHNO 2014), IARIA, pp. 1-7, 2014.

The latter publication has been awarded to submit an extended article of this publication to the IARIA Journal on Advances in Life Sciences:

Paper 6
S. Rosenthal and M. Borschbach, „Impact of Population Size, Selection and Multi-Parent Recombination within a Customized NSGA-II for Biochemical Optimization“, *International Journal on Advances in Life Sciences*, IARIA, vol. 6, nr. 3&4, pp. 310-324, 2014.

Paper 7
S. Rosenthal and M. Borschbach, „A Concept for Real-Valued Multi-Objective Landscape Analysis Realized on Biochemical Optimization Problems“, *Procee-*

dings of the 18th European Conference on the Applications of Evolutionary Computation (EvoApplications 2015), LNCS 9028, pp. 897-909, 2015.

Paper 8
S. Rosenthal, B. Freisleben and M. Borschbach „Aggregate Selection in Multi-Objective Biochemical Optimization via the Average Cuboid Volume Indicator“, Emmerich et al. (Eds.): *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation VI*, 2015, to appear.

Paper 9
S. Rosenthal and M. Borschbach „Average Cuboid Volume as a Convergence Indicator and Selection Criterion for Multi-Objective Biochemical Optimization“. Emmerich et al. (Eds.): *EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation VII*, DOI 10.1007/978-3-319-49325-1_9, 2017.

The experiments presented in the publications (1) to (6) have been performed with the software tool developed in the BMBF project OPTOPROBE within the research laboratory Optimized Systems of the University of Applied Sciences FHDW in Bergisch Gladbach under the supervision of PD Dr. Markus Borschbach. This project is a collaborative project of the research group of PD Dr. Andreas Frey at the Leibniz Research Center Borstel and several research partners. The foundation of the developed evolutionary platform roots back to several earlier research cooperations with the same research group of PD Dr. Andreas Frey and PD Dr. Markus Borschbach while being a visiting Professor at the Technical University in Chemitz and at the Institute of Computer Science in Münster. The experiments presented in (8) and the experimental results presented in this thesis have been performed with the extension of the open source tool jMetal.
The author of this thesis has developed all main contributions of the papers listed above, performed the experiments, wrote the publications and presented the publications 2, 3, 4, 5 and 7 at the different conferences. The developed components of the publications have also been included in the OPTOPROBE software tool by the author of this thesis as well as in the extension of the tool jMetal.
The theoretical part of the publications 1 and 2 are included in Sections 4.5 and 4.4. The theoretical part of the publications 3, 4 and 8 are included in

Section 4.6. The theoretical and experimental landscape analysis parts of the publications 6, 7 and 9 are included in Section 3. The cost-utility analysis of the open software tools in Section 5.1 has been partly published in the publications 5 and 6.

1.6 Organization

This thesis is organized as follows. Chapter 2 presents a summary of related work regarding established MOEA approaches as well as an overview of related research in the field of MOEA adaptation for the final aim of peptide optimization. Furthermore, the biochemical objective functions constituting the three- and four-dimensional multi-objective molecular optimization problems are presented in this chapter.
Chapter 3 presents a review of existing qualitative and quantitative techniques for landscape analysis. For a deep insight into the problem complexity of the proposed three- and four-dimensional molecular optimization problem, this chapter proposes a landscape analysis of the multi-objective molecular landscapes based an analysis concept in particular applicable for real-valued Multi-objective Molecular Landscapes (MOML).
Chapter 4 describes the customized NSGA-II for peptide optimization - termed as VONSEA - in detail. At the beginning of this chapter, the traditional procedure of NSGA-II is presented, followed by the introduction and the motivation of the VONSEA components. Thereby, a thorough review of the traditional operators for each component (recombination, mutation, selection) is presented, including a classification of these operators. The VONSEA components are further classified by these categories.
Chapter 5 introduces the indicators for the EA objectives convergence, diversity and relative non-dominated solution quality. The established convergence and diversity indicators for the assessment of MOEA performance are presented, and a new convergence indicator is introduced and its properties are discussed.
Chapter 6 presents the experimental results of VONSEA in the case of the three- and four-dimensional MOP with the focus on the hypotheses and questions raised above.
Chapter 7 summarizes the results, discusses the hypotheses and questions and gives an outlook of future research directions.

2 Related Work

2.1 Multi-objective Evolutionary Algorithms

EAs are heuristic methods that are categorized into three research areas: The biggest and most popular part is the area of GA beneath the areas of Evolutionary Strategy (ES) and Evolutionary Programming (EP). In the 1970s, GA and ES were developed in parallel by Holland [84] in the case of GA and by Rechenberg [127] in the case of ES. The three categories have commonalities as well as differences: GA and ES operate on fixed length strings. However, GAs traditionally make use of a bit string encoding, whereas ESs prefer real-value vectors as ESs have once been designed for parameter optimization problems. Furthermore, all categories of EAs incorporate a mutation operator, GAs and ESs additionally use recombination operators. All algorithms of the EA areas make use of selection operators. GAs prefer probabilistic operators for parent selection, whereas ESs use deterministic selection strategies to choose the individuals for the succeeding population. Another difference between GA and ES is the population size, GAs operate with fixed population sizes, whereas the population size within ESs is variable. Furthermore, it is established in the GA research area to evolve algorithm with a problem specific coding according to the optimization problem. The problem-specific encoding is not common in the area of ES. A more detailed comparison of these research areas is described in related work [89].
The terminology EA is mainly used throughout this thesis and refers to the abstract commonalities of ES and GA. The terms ES and GA are used in the context of metaheuristics as related work.
The first MOGA was proposed by Schaffer in 1985 [136] and is an extension of the simple GA. This algorithm is termed Vector-Evaluated Genetic Algorithm (VEGA). A special selection process is used in VEGA based on proportional selection and the objective switching rule: The population of the size N is divided into q subpopulations P_i of the size N/q, whereas q is the number of objective functions. These subpopulations are created by performing pro-

portional selection according to each objective function z_i (for $i = 1, ..., q$). The subpopulations P_i are then stored together into a mating pool of size N. This mating pool is shuffled and crossover as well as mutation are applied. The crossover and mutation operators are like the one in the simple GA. The disadvantage of VEGA is a consequence of the selection process: the solutions tend to converge to the optimum of each objective function.

From 1985 to 1997 the method of weighted objectives was predominant in the area of MOO. Multiple objectives are combined into one single-objective scalar function via a weight vector. The main difficulty arose in the determination of the optimal pre-defined weights. In 1993, Fonseca and Fleming proposed the first GA which explicitly uses a Pareto dominance scheme for comparison of the solutions (Pareto-based ranking) and niching techniques, termed Multi-Objective Genetic Algorithm (MOGA) [65]. The motivation of MOGA is an acceleration of the search process towards the true Pareto front while maintaining diversity within the solutions. In order to fulfill these requirements, a fitness sharing is only used between solutions with the same Pareto rank. Niching distance measures of two solutions are determined to compare them with a sharing parameter. If the distance is less than the sharing parameter, the associated niche count of the solution is adjusted. Unfortunately, MOGA usually provides slow convergence and the performance depends on the sharing parameter. An improved version of MOGA was introduced and benchmarked in 2001 by Purshouse and Fleming [124].

In 1994, Horn, Nafpliotis and Goldberg presented the Niched Pareto Genetic Algorithm (NPGA), which introduces the Pareto domination tournament and the class fitness sharing [87]. The selection is realized by a tournament selection scheme based on Pareto dominance [76]. Two solutions are chosen by binary tournament selection. Both are compared to a tournament set (usually of a size of 10% from the population) randomly chosen from the entire population. If only one of both solutions is dominated by the tournament set, then the non-dominated one is selected. If both solutions are either dominated or non-dominated, the selection is decided by class fitness sharing. Thereby, the solution in the less crowded niche is selected. The disadvantage of NPGA is its performance dependence on the niche parameter and the size of the tournament set. An improved version of NPGA termed NPGA-2 was published in 2001 by Erickson et al. [57].

Hajela and Lin introduced the Weight-Based Genetic Algorithm (WBGA) at the beginning of the nineties [79], which is related to VEGA. WBGA transforms the MOP into a scalar optimization problem by multiplying each objective function with a weight. These weights are not fixed and encoded in the

individual vectors. The fitness values of a solution are calculated by adding the weighted objective function values. To maintain diversity, subpopulations of individuals are evaluated for different objectives analogous to VEGA. Multiple solutions are simultaneously searched in a single run. A niching method is used on the weight vectors to maintain diversity.
The Random-Weighted Genetic Algorithm (RWGA) introduced by Murata and Ishibuchi in the middle of the nineties [114], [97] is also based on a weighted sum of objective functions, but the weights are assigned randomly to allow a flexible search direction over the search space. Elite solutions are maintained by elitism strategy [97]. Non-dominated solutions are stored in an external population, which is distinguished from the current population. Elitism is realized by selecting a specific number of the external set for the succeeding generation. RWGA as well as WBGA have difficulties in finding solutions uniformly distributed over non-convex true Pareto fronts.
Scrinivas and Deb present one of the mostly used MOGAs, the Non-dominated Sorting Genetic Algorithm (NSGA) in 1994 [142]. NSGA is built upon the basic framework introduced by Holland [84]. The innovation of NSGA is the selection procedure: the stochastic remainder proportional selection (SRS). This procedure is implemented to have a good convergence of the solutions to the Pareto front. The ranking is based on the Pareto dominance principle. More precisely, solutions are assigned to a dummy fitness value for each dominance class, proportional to the population size. Fitness sharing within each class is incorporated to ensure diversity within the solutions. NSGA is criticized for the high computational complexity $O(MN^3)$ (where M is the number of objectives and N the population size), its lack of elitism and the challenge of determining the sharing parameter. An improved version of NSGA including elitism was published in 2001 by Deb [37].
An improved version of NSGA was proposed in 2002 by Deb and is termed NSGA-II [44]. The computational complexity for the used fast non-dominated sorting is reduced to $O(MN^2)$. The selection procedure is based in the solutions rank and the crowding distance operator. Elitism is included and a sharing parameter is implemented, which is not chosen *a priori*. A more detailed description of NSGA-II is given in a later chapter.
In 1999, Zitzler et al. proposed the elitist evolutionary algorithm termed Strength Pareto Evolutionary Algorithm (SPEA) [185]. SPEA assigns better fitness values to non-dominated solutions in less crowded parts of the objective space. Therefore, an external population of a fixed size stores all non-dominated solutions up to the actual generation. The individuals for the succeeding generation are selected from the current and an external set. The selection probability of

an individual depends on a strength value that reflects the number of individuals dominated by or equal to this individual. To ensure diversity among the non-dominated solutions, a deterministic clustering technique is used, but it does not include a distance measure between solutions. The point of criticism is the high complexity of the clustering algorithm. Furthermore, SPEA does not converge to the Pareto front in the case that the front is concave as the used fitness assignment is sensitive to concave fronts. Zitzler et al. published an improved variant of SPEA termed SPEA2 in 2001 [183]. SPEA2 differs to SPEA in the fitness assignment and the diversity mechanism. A density measure based on the k-th neighbors is used as diversity mechanism. The clustering technique of SPEA is replaced by a truncation method in SPEA2.
Knowles and Corne proposed a simple (1+1) local evolutionary algorithm termed Pareto Archived Evolutionary Strategy (PAES) in 2000 [95] that uses an archive pool for selection and a hypergrid strategy. A solution is randomly generated at the beginning of the algorithm and stored in the archive pool. After that, another solution is generated by mutating the initial one. The mutant is compared to the initial solution and only accepted as new parent if it is non-dominated, otherwise it is discarded. In the case that both are indifferent, the archive is used for comparison. An archive pool of fixed size is used to store all non-dominated solutions. The objective space is divided by a grid and the areas are evaluated with regard to its crowdedness. If the newly generated solution does not dominate any other solution in the pool, the decision for the current solution or the entry in the pool is done by crowding measure. In the case that a non-dominated solution enters the archive pool, solutions dominated by this one are deleted.
In 2001, Corne und Knowles introduced the Pareto-Envelope-based Selection Algorithm (PESA) [32] that uses principles of SPEA and PAES. To ensure a high diversity within the solutions, the objective space is divided into k-dimensional hyper-boxes. The characteristic feature of PESA is the selection method. The aim of this selection is that non-dominated solutions in less crowded boxes have a higher chance to be selected. The selection probability of an individual depends on a squeeze factor that is the number of individuals sharing the same box. Binary tournament selection is used and the individual with the lowest squeeze factor is chosen. An improved version of PESA termed PESA-II was published by Corne et.al. in 2001 [31]. The difference to PESA is the replacement of the individual-based selection. A selective fitness is assigned to all solutions in a hyper-box and the whole hyper-box is selected. Solutions are chosen randomly as parents of this hyper-box. The motivation for PESA-II is the fact that hyper-box selection ensures a better spread of

the solutions than individual-based selection. *A priori* information about the objective space is mentioned as a disadvantage of these algorithms.
In 2003, Deb proposed a steady-state MOEA based on the ϵ-concept of Laumann [101] termed ϵ-MOEA [43]. The motivation for this algorithm is a good compromise between convergence, a high diversity of the solutions and a low computation time. ϵ-MOEA comprises a parent and archive update strategy. An offspring is created from a solution of the parent and one of the archive pool. Both offspring are used to update the parent pool and archive pool. The usual Pareto dominance principle is used to update the parent population and the ϵ-dominance principle is used to update the archive pool. In the sense of the ϵ-dominance, two solutions which have a difference less than ϵ_i in the i-th objective are not non-dominated to each other.
In 2003, the Rank-Density based multi-objective Genetic Algorithm (RDGA) was proposed by Lu and Yen [107]. This algorithm is characterized by a special ranking method - Automatic Accumulated Ranking Strategy (AARS), a rank-density based fitness assignment and a forbidden region concept. The rank value of a solution is determined as the summation of the rank values of the solutions which are dominated by the first one plus 1. The value 1 is the assigned rank value for non-dominated solutions. To maintain diversity, a modified cell density evaluation scheme like the one in PESA is used. The rank and the density are two features calculated for each solution. Then, a modified VEGA is applied to fulfill the fitness assignment. A forbidden region concept is implemented to prevent the 'backward' effect. The forbidden regions include all cells dominated by the selected parents for crossover and mutation. Offspring from these cells will not take part in the succeeding generation.
In 2004, Zitzler and Künzli introduced the Indicator-based Evolutionary Algorithm (IBEA) [182]. Practically, it is of the same procedure as NSGA-II and SPEA2, but it differs in the selection process and it uses one population of variable size (instead of a fixed actual population and an archive of the best solutions like in SPEA2). The main goal of the selection process is the dominance preserving by a binary quality measure (indicator) which is used within the selection process. In IBEA, the fitness of the individuals is determined according to the value of the binary quality measure. Within the binary tournament selection only pairs of individuals are compared and the worst are removed from the population. IBEA does not make use of a crowding or density concept for diversity preserving like in NSGA-II or SPEA2.
In 2005, Emmerich et al. [54], [17] proposed the S-metric Selection Evolutionary MO Algorithm (SMS-EMOA). This algorithm is a steady-state EMOA that combines the concept of non-dominated sorting with a selection operator based

on the S-metric. SMS-EMOA is similar to NSGA-II except for the selection procedure and the ranking. In each iteration, only one individual is generated by variation operators. The selection of SMS-EMOA chooses the subset of individuals for the succeeding population according to the subset with the maximal S-metric under all possible subsets. The disadvantage of this algorithm is the computational complexity for the calculation of the hypervolume, especially if the number of objectives is greater than 3. Otherwise, SMS-EMOA features convergence behavior to the Pareto front even in the case of high dimensional solution space - where NSGA-II and SPEA2 failed [167].
In 2007, Zhang and Li presented the MOEA using Decomposition (MOEA/D) [178]. The concept of MOEA/D is to composite the MOP into different subproblems that are solved simultaneously. The objective of each sub-problem is an aggregation of several objectives. A method for converting an MOP into several scalar optimization problems is for example the weighted sum approach. Each subproblem is solved by using information about the neighbored subproblems: neighbored subproblems are defined on the basis of the distances between the coefficient vectors obtained by the aggregation. Consequently, the population in each generation is composed of the best solutions found so far for each subproblem. This algorithm is of lower computational complexity than NSGA-II. A short overview of the characteristic features of the most traditional MOGAs is given in Table 2.1 and a further short description as well as a comparative overview of the established MOEAs is available in other work [97], [72].

2.2 Multi-objective Evolutionary Peptide Optimization

Molecule optimization formulated as optimization problems usually provides several conflicting objectives. Therefore, MOEAs have become established methods in the field of peptide- or protein-based drug design. Some work has been published in the recent years regarding the use and the adaptations of state-of-the-art MOEAs for the purpose of multi-objective molecular optimization. This section gives an overview of this work published so far.
Cutello, Narzisi and Nicosia presented a more sophisticated version of the $(1+1)$ local search evolutionary strategy PAES to predict the native structure of a protein from the amino acid sequence [33]. The popular protein structure prediction problem is reformulated as a 2-objective optimization problem by

	Fitness assign.			Diversity mechanism						selection strategy							Archive	Elitism	steady-state
	Dominance based	scalarization based	objective based	objective based	fitness sharing	niching technique	cell-based density	crowding distance	clustering	forbidden region concept	objective-proportionate	SUS	tournament	Pareto-based	proportionate (prop.)	stochastic remainder prop.	indicator-based			
VEGA			x								x									
MOGA	x				x	x						x		x						
NPGA	x				x	x							x	x						
WBGA		x			x	x									x					
RWGA		x											x	x	x			x		
NSGA	x				x	x								x		x				
SPEA	x				x				x				x	x				x		
PAES	x						x							x				x		
PESA	x						x						x							
NSGA-II	x							x					x	x						
ϵ-MOEA	x						x						x	x				x		
RDGA	x						x			x										
IBEA													x				x			
SMS-EMOA																	x			
MOEA-D		x					x													

Table 2.1: Overview of the reviewed state-of-the-art MOEAs according to their main characteristic techniques.

decomposing the CHARMM energy function. The test runs are performed on a set of medium to large proteins with 26 to 70 amino acids. First test runs with the traditional PAES algorithm revealed poor performance caused by a premature convergence and a trapping in local minima. The more sophisticated version I-PAES makes use of polypeptide chains as solution representations, a cloning, a hypermutation and a selection strategy based on the Pareto dominance principle. Two clones are produced and both are mutated by the hypermutation. The first mutation changes the conformation more drastically and the second mutation ensures a more local search. The mutation probabilities are determined by an exponential function, which decreases as the search method proceeds. Then, the non-dominated clone serves as a new mutated solution and the other one is added to the archive. After that, the standard procedure of PAES follows.

Hohm, Limbourg and Hoffmann published a MOEA for the design of effective peptide-based drugs [83]. This MOEA is applied on a 3-objective optimization problem referring to the mimic antibody epitopes of the proteins thrombin and blood coagulation factor VIII as first objective, short peptide sequence as second objective and conformationally stable peptides as third objective. The procedure of the proposed MOEA starts with the initialization of a population of sequences. Mutation, crossover and swapping are used as variation operators and a three-criteria based selection strategy is used to include the idea of elitism as well as genetic diversity. The mutation process makes use of a mutation pool comprising single amino acids for mutation as well as short amino acid sequences. Therefore, an amino acid is replaced by another amino acids or by an insertion of a short sequence. Also the mutation pool undergoes a selection process: a fitness value is assigned to the members of this pool according to the number of times they have been chosen as well as the number of times they have been successful. Pairwise single mutation crossover is used and the crossover points are chosen randomly. Since molecules sometimes provide good motifs, but in a suboptimal ordering, the motives are swapped. The selection process makes use of the idea of elitism ensuring that the best individuals are not lost. Therefore, a fixed number of the best individuals from the archive find their way into the succeeding generation. Furthermore, binary tournament is applied to select the remaining individuals of the succeeding generation. The individuals are selected based on three criteria:

1. Pareto dominance principle,

2. Diversity preserving strategy realized by assigning a value to each solu-

tion based on the number of individuals sharing the same hypercube,

3. Principle of change realized by the random selection of an individual.

The test runs are performed with a population size of 10. The mutation pool consists of 8 motifs and 20 amino acids.
Oduguwa, Tiwari, Fiorentino and Roy use three different MOEAs to determine a good protein-ligand configuration for a given target protein and its binding components [118]. The three algorithms PAES, SPEA and NSGA-II are investigated regarding their drug candidate discovery abilities for the protein-ligand docking problem. The framework including these three algorithms makes use of a specific chromosome structure comprising three coordinates of the chromosome in the target axes system, two angles of the chromosome compared to the reference compound and a set of relative coordinates of the chromosome in the compound axes system. PAES, SPEA and NSGA-II are compared to each other in solving a 3-objective MOP comprising the internal energy of the compound, the protein-compound couple' s Van der Waals and electrostatic energy of interaction as well as the shape complementaries. The population size was set to 100, and 500 generations were performed. NSGA-II and PAES performed best, but the optimal solutions were found by all three MOEAs.
Lee, Shin and Zhang published the NSGA-II with constrained tournament selection for the DNA sequence optimization [104]. The DNA sequence problem is formulated as a 4-objective MOP with two constraints. The constraints are the number of bases G and C and the melting temperature. This specific NSGA-II uses a two-stage crossover process. The first stage is a sequence set level crossover, which is performed by an exchange of the sequences between two chromosomes. The second step is the one-point crossover. Furthermore, the one-point mutation is used on every chromosome. The constrained tournament selection favors solutions, which are feasible, have less penalty or belong to a better front. Therefore, the selection process comprises three cases: Firstly, the feasible solutions are selected, secondly the one with less penalty is selected and thirdly the dominating one is selected or otherwise the one with the larger crowding distance. The sum of penalties is used for each constraint as the penalty of the chromosome. The experiments were performed with a population size of 1000 and 200 generations.
Rajapakse, Schmidt and Brusic presented a work using the NSGA-II to search for a motif that unravels rules governing peptide binding to medically important receptors in the application field of drug design and vaccines target discovery [125]. The NSGA-II makes use of a chromosome presentation by an

ensemble of $k \cdot n$ real numbers, where k presents the motif length and n the number of residues. The performance is investigated on a 2-objective MOP with a population size of 500 and 300 performed generations. NSGA-II is superior to other computational techniques.
Gilles, Willet, Flemming and Green proposed the program MoSELECT that includes MOGA [65] for the purpose of combinatorial library optimization regarding different properties like diversity and 'drug-like' physiochemical properties [73]. In a further work, MoSELECT has been used to optimize the library size and the configuration of the combinatorial libraries [174].
Deb and Reddy published experiments on three NSGA-II variants for the identification of the optimal gene subsets for the three commonly used cancer data sets Leukemia, Lymphoma and Colon [45]. The traditional NSGA-II as well as the two alternatives are applied on a 3-objective MOP, where the objectives refer to the gene subset size, the number of misclassified training samples and the number of misclassified test samples with a population size of 500 and a performed generation number of 500. The alternative NSGA-II makes use of the biased dominance principle referring to the objective i. The biased dominance principle ensures that two solutions with identical complementary objective values j and $j \neq i$ are not dominating each other. Furthermore, solutions lying along the f_i axis have the potential to be non-dominated to each other. The third provided alternative NSGA-II is the multimodal NSGA-II. Solutions that are equal in the objective space but have different phenotypes are termed multimodal solutions. The solutions providing identical classifications are of special interest in the field of biology. The selection process determining the succeeding generation is modified: The selection of the solutions proceeds like in the traditional NSGA-II until the last front. If the number of distinct solutions (distinct solutions are differing in at least one objective function) is higher than the number of solutions required to fill the succeeding population, the crowding distance measure is used as criterion. Otherwise, a procedure is used that fills the population with a proportional number of multi-modal solutions of every distinct solution corresponding to their appearance in the last front.

3 Multi-objective Molecular Landscape Analysis

Fitness landscape analysis is commonly used to gain an insight into the difficulties and complexity of an optimization problem as well as to provide the opportunity to predict the ability of heuristic search algorithm in finding considerably good solutions [112]. The use of MOEAs for molecule optimization has increased significantly, but the general understanding of the molecular landscape properties with the aim of designing an appropriate MOEA to search the molecular space is missing [55]. The analysis of landscape structures provides information about landscape difficulties of molecular optimization problems. This information provides a better insight into the composition of a MOEA with optimized search performance regarding a particular type of algorithm, the types of variation probabilities as well as the selection pressure for a suitable balance of global and local search behavior. The components of a fitness landscape are a set of genotypes, the fitness functions, which evaluate the genotypes and the genetic operators, which represent the move operator for the exploration of the neighborhood. Stadler presented the formal description of landscape composition [149]:

Definition 6 *A landscape consists of three ingredients:*

- *A set X of configurations,*
- *a notation X of the neighborhood, the nearness, distances or accessibility on X, and*
- *a fitness function $f : X \to \mathbb{R}$.*

The local optima as the fundamental characteristics are defined as:

Definition 7 *The space X is assumed to be metric and $x^* \in X$ is a local*

maximum (minimum) if there exists $\epsilon > 0$ *such that*

$$f(x^*) \geq f(x) \quad or \quad (f(x^*) \leq f(x)) \tag{3.1}$$

for all x *in the neighborhood of* x^**:* $x \in N_\epsilon(x^*)$.

Global optima are defined as the absolute maxima or minima of the search space.
The main goal of landscape analysis is to determine landscape characteristics that state the structure of the landscape and have a strong influence on the heuristic search performance [103], [55]: modality[1], correlation[2], ruggedness[3] and plateaus[4].
A concrete landscape analysis starts by specifying metrics that characterize the geometric properties. The selection of suitable metrics depends on the organization of the configuration space X and has to take account of the optimization problem. Reidys and Stadler [129] summarized three distinct approaches for the organization of the configuration space X:

1. Transition probabilities are used to describe the movement from one configuration to another. The process is describable by Markov chains and is especially applied in the case of combinatorial optimization problems.

2. In the field of computer science, genetic operators are usually used as move operators to create new solutions.

3. Rigorous mathematical analysis is performed via specified metrics or topologies on X.

The landscape analysis used in this thesis is oriented on the analysis of physiochemical functions predicting peptide properties. According to Definition 6, the three ingredients are assigned as follows: the configuration set X consists of all feasible peptides with a length of 20 composed of 20 amino acids. Referring to the three approaches of the configuration space organization, the use of Markov chains is not advisable caused by the general difficulty to efficiently design highly complex spaces [28], especially for the highly complex

[1] Modality is a feature that provides an overview of the tendency of the fitness landscape to produce local optima.
[2] Correlation describes the dependence between two solutions.
[3] Ruggedness is a characteristic of the landscape for fitness variation between the fitness values of a solution and its neighbored points.
[4] A plateau is a feature that represents neutrality referring to a solution set with equal fitness.

space X. Furthermore, the configuration set allows no mathematical definitions of metrics or topologies. Consequently, the organization of such a biochemical landscape is based on the second approach. According to the second ingredient, a genetic operator is used to explore the neighborhood of a configuration, as proposed by several authors [103], [55], [111]. According to the third ingredient, biochemical fitness functions usually compose discrete[5] search spaces as there are real-valued solutions which have no corresponding feasible peptide in the search space. The fitness functions composing the three- and four-dimensional MOP used as benchmark problems in this thesis and presented in section 3.4 are discrete.
The analysis techniques described in the following section are common for landscape analysis performed on sequences of solutions obtained by random walks.

3.1 Single-objective Fitness Landscape Analysis

Different techniques have been introduced to analyze the characteristics of single-objective fitness landscapes. These techniques are divided into two categories: Statistical analysis and information analysis [110]. In the case of both, statistic and information analysis, the fitness landscapes are considered as statistically isotropic[6] [161].

3.1.1 Statistical Analysis

Statistical analysis comprises different correlation metrics to estimate the structure of a landscape. The autocorrelation function and the correlation length are two established measures of the category statistical analysis. The autocorrelation function measures the ruggedness of the landscape and was introduced by Weinberg [170]. The autocorrelation function p is defined as:

$$p(d) = \frac{< (f(x) - < f >) \cdot (f(y) - < f >) >_{d(x,y)=d}}{var(f)}, \tag{3.2}$$

[5] **Definition of a discrete set:**
X is a topological space and the set $S \subset X$. S is said to be discrete if every point $x \in S$ has a neighborhood U such that $S \cap U = x$.

[6] The term isotropic refers to the fact that the statistics of the time series $\{f_i\}$, resulting of the random walks, are the same and independent of the starting point. Concluding, this means that the landscape has globally the same structure everywhere [88].

where $< x >= \frac{1}{N} \cdot \sum_{i}^{N} x_i$ denotes the average value of all x_i and $var(f)$ the variance. This measure defines the correlation of solution points at the distance d in the search space. Weinberg also proposed an alternative, the random walk correlation function. Thereby, $r(s)$ is used as an autocorrelation function along a random walk [170]:

$$r(s) = \frac{< (f(x_i) - < f >) \cdot (f(x_{i+s}) - < f >) >}{var(f)}, \tag{3.3}$$

where $\{f(x_i)\}$ are the time series containing the fitness values of the random walk steps $\{x_i\}$. This correlation function calculates the correlation between two solution points with a distance step length of s on the random walk path. In general, $r(s)$ starts with a values of 1 for $s = 0$ and results in $r(s) = 0$ for a distance step $s > 0$. The faster the descent of $r(s)$, the more rugged is the landscape.

On the basis of these random walk correlation functions, another established measure of the statistical analysis is defined. The correlation length defines the distance beyond which two solution sets become uncorrelated:

$$l = -\frac{1}{ln(|r(1)|)}, \tag{3.4}$$

for $r(1) \neq 0$. For interpretation, the higher the correlation length, the smoother is the landscape. The smoother a landscape, the higher is the correlation of neighbored solutions and the search process is less challenging for the search heuristic [112].

Fitness Distance Correlation (FDC) was proposed by Jones [90] as an analyzing technique to detect search difficulties with the aim of examining GA performance on optimization problems with known optima. The FDC coefficient measures the relation of the fitness and the distance of the solutions $\{s_i\}$ to the nearest optimum x^* in the search space:

$$FDC = \frac{cov(f(s_i); d(s_i))}{\sqrt{var(f(s_i)) \cdot var(d(s_i))}}, \tag{3.5}$$

where d is the distance function to x^* and $cov(x; y)$ is the covariance. The coefficient values are in the interval $[-1; 1]$. Jones further introduced three categories to classify the FDC coefficients according to the prediction of GA effectiveness in solving optimization problems:

- $FDC \geq 0.15$: The fitness increases with the distance. The GA is potentially not effective or the problem is misleading.

- $-0.15 < FDC < 0.15$: There is virtually no correlation between fitness and distance. The problem is categorized as difficult.
- $FDC \leq -0.15$: The fitness increases as the optimum approaches. The GA is potentially effective or the problem is straightforward.

A great disadvantage of FDC is that the nearest optimum or at least the best known solution has to be known in advance.

3.1.2 Information Analysis

Information analysis from a global perspective is a quantitative landscape analysis. The aim of information analysis is to quantify the characteristics of the landscape like modality, ruggedness and regularity degree by the size, form and distribution [161]. Therefore, Vassilev et al. [161] introduced three threshold-based indicators analyzing the structure of a fitness landscape: these indicators are termed Information Content, Partial Information Content and Information Stability and are based on the landscape path $\{f_t\}$ containing the fitness values in form of real numbers that are obtained by random walks. f_t is the fitness value of the genotype x_t achieved at step t from the start point. The path $\{f_t\}$ is transformed into a string $S(\epsilon) = s_1 s_2 ... s_n$ with $s_i \in \{-1, 1, 0\}$, where

$$s_i = \begin{cases} -1 & \text{if } f_i - f_{i-1} < -\epsilon \\ 1 & \text{if } f_i - f_{i-1} > \epsilon \\ 0 & \text{if } |f_i - f_{i-1}| \leq \epsilon \end{cases} \tag{3.6}$$

and $\epsilon \in [0; l]$, where l is the maximal difference between two fitness values. The indicator is more sensitive to movements of the random walk the smaller the value for ϵ.
The Information Content is an entropic measure and defined via:

$$H(\epsilon) = -\sum_{p \neq p} P_{[pq]} log_6(P_{[pq]}), \tag{3.7}$$

where $p, q \in \{-1, 1, 0\}$, $P_{[pq]} = \frac{n_{[pq]}}{n}$ are the probabilities presenting frequencies of possible blocks pq and $n_{[pq]}$ is the number of occurrences of the blocks pq in $S(\epsilon)$. The base of the logarithm is chosen as 6. This is the number of all possible blocks pq. This indicator measures the ruggedness of the landscape path. Thereby, the variation of parameter ϵ has the effect of a more detailed or a more global look on the landscape path.

For the determination of the Partial Information Content, a string $S'(\epsilon)$ is constructed of $S(\epsilon)$ by deleting the elements 0, and consecutive equal elements are reduced to one element of the string $S(\epsilon)$. The partial information content is defined as:

$$M(\epsilon) = \frac{v(\epsilon)}{n}, \tag{3.8}$$

where $v(\epsilon)$ is the length of $S'(\epsilon)$ and n the length of $S(\epsilon)$. Furthermore, $v(\epsilon)$ indicates the number of extrema along the landscape path. In the case of $M(\epsilon) = 0$, the landscape path is nearly flat or monotonously increasing or decreasing. Otherwise, $M(\epsilon) = 1$ indicates that the landscape path is maximal rugged.

Information Stability as the third indicator for information analysis is proposed by Vassilev. It is an indicator for the highest difference between neighboring points in the landscape path. The information stability is defined as the smallest value of ϵ for which the landscape path becomes flat. In this case, the string $S(\epsilon)$ comprises only zeros.

Another information indicator was proposed by Leier et al. [105]. This indicator gives information about the basin[7] density as well as length of flat areas. Therefore, it is an indicator for the ratio between flat and smooth parts of a landscape path and therefore an optimal measure for neutrality. It is defined as:

$$h(\epsilon) = -\sum_{p \in \{-1,1,0\}} P_{[pp]} log_3(P_{[pp]}), \tag{3.9}$$

where $P_{[pp]}$ is the frequency of blocks pp in $S(\epsilon)$.

3.2 Related Work on Molecular and Multi-objective Landscape Analysis

The aim of molecular landscape analysis is to gain an insight into the hypothesis: the similarity in the molecule structure is often related to the similarity in its molecule properties [55]. More generally, molecular landscape analysis provides an understanding of molecular search space properties, which is necessary to design an appropriate search algorithm with regard to the landscape

[7] **Definition: Basin of attraction** [128]
Basin of attraction of an optimum x^0 is the set $B(x^0) = \{x \in X : \mu(x) = x^0\}$, where $\mu : X \to X^0$ is the neighborhood search. If $x \in X$ is an initial point, $\mu(x)$ is the nearby optimum that x reaches. The size of the basin of attraction of a local optimum x^i is the cardinality of $B(x^i)$. The basin size is also used as a measure for ruggedness.

structure.

In previous work [103], four molecular fitness landscapes are investigated regarding the properties of modality, ruggedness, neutrality, local optima and basins via correlation measures and information analysis. The essential components of a molecular landscape are the representatives of the peptides (genotypes) and the fitness functions. The search space is given by the set of all feasible peptides of a length of 5 comprising 23 amino acids. The analysis of the molecular landscapes Oestrogen Receptor, Peptide Receptor, Lipoxygenase Inhibitor and Neuropeptide Y_2 Receptor is performed on the basis of random walks. The solutions are encoded as character strings and the initial solutions of the random walks are determined randomly. Taking account of this genotype representation, the random walk is performed by mutation as move operator, which explores the neighborhood of a solution. The neighborhood of a peptide is defined by the peptides differing in one amino acids at any position of the peptide. Therefore, a new solution or the succeeding time series step of the random walk is generated by mutating one amino acid at any position. The performed random walks are evaluated by a correlation measure to reveal the ruggedness of the landscape. As the molecular search space is of a high complexity (23^5 feasible peptides), the random walks of a length of 100 are relatively small compared to the complexity. Consequently, the average values and the standard deviations among the various time series steps are usually time varying instead of stationary and this phenomena is termed time-varying volatility. In a more abstract term, the variance of the fitness function values achieved by a random walk changes over the random walk steps. To overcome this problem, the autocorrelation of Weinberg (eq. (2.31)) is adapted by determining the average value and the standard deviation over all configurations of the random walks and the average value is applied on the starting point to prevent high errors in estimation:

$$p_s = \frac{\frac{1}{n+1}\sum_{i=0}^{n}(x_{i0}-\mu)(x_{is}-\mu)}{\sigma^2}, \tag{3.10}$$

where μ is the average value calculated by

$$\mu = \frac{1}{n}\sum_{i=1}^{n} f(x_i), \tag{3.11}$$

σ is the standard deviation determined by

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-\mu)^2} \tag{3.12}$$

and x_{i0} is the starting point of the i-th random walk. The self-correlation value of the starting point p_0 has to be 1 or at least approximately 1. Furthermore, information content (eq. 3.7), partial information content (eq. 3.8) and neutrality measure (eq. 3.9) are applied to analyze the structure of the molecular landscape [103].

Regarding the used analysis techniques, Lee [103] provides a traditional landscape analysis for each of the four molecular fitness landscapes. An analysis concept for MOML is still an open research task to date. Garrett and Dasgupta [70] however provide a multi-objective fitness landscape analysis according to local and Pareto optima, fitness distance correlation and ruggedness in the form of a high-level overview and discussion. A number of potential analysis methods are described and discussed to analyze these characteristics for single- and multi-objective fitness landscapes. The following resume refers to the statements of the multi-objective landscape analysis [70]:

Information about the landscape are intuitively obtained by the distribution and the number of local and Pareto optima. The number and distribution of local optima have a significant impact on the performance of a general-purpose search algorithm. The higher the number of local optima, the higher the probability that the search algorithm traps in a local optimum. Furthermore, the distribution of local optima has also a strong impact on the search algorithm performance. The common example is the highly differing optima distribution of the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP): TSP reveals a Big-Valley-Structure, where all local optima are positioned around a line that approaches the global optima. Such a specific structure provides the opportunity to be exploited by the search algorithm. Otherwise, the local optima in the case of QAP are nearly uniformly distributed over the search space, but reveal no certain structure such as in the case of the TSP landscape. Therefore, the detection of a single local optima provides no information about how to reach the other local optima or even the global optima. Consequently, QAP is more challenging for a search algorithm than the TSP.

In the sense of single-objective optimization, FDC is the correlation coefficient between the distance in the objective and the distance in the solution space. This correlation is commonly visualized by scatter plots of objective distances and distances in the solution space. Ideally, the changes on a solution required to move from a local to a global optima have to be minor, Otherwise a search algorithm potentially fails in finding the optimal solution. In the case of multi-objective optimization, global optima are referred to as non-dominated solutions. Many of the Pareto optimal solutions are also global

optima of different single-objective functions. Thus, the correlation between the non-dominated solutions is not necessarily resembling to the correlation between different local optima of a single-objective function. The correlation between non-dominated solutions reveals important information about the difficulty to exploit the Pareto front. FDC requires the mapping of distances to the optima into a single distance value. Therefore, Garret proposes two potential distance measures: The Euclidean distance between the fitness vectors is a potential metric for this purpose. Another proposed possibility is to define the angles between these vectors as 'distances'. These analysis techniques and the impact on the analysis have not been investigated theoretically or empirically so far.

Garrett defines ruggedness of landscapes intuitively: Many local optima of highly varying fitness values are positioned in any region of the landscape. Different techniques for the investigation of the landscape ruggedness are established in the case of single-objectives problems. The distribution and the number of local and Pareto optima are indicators for ruggedness. The correlation between adjacent solutions is another method to detect ruggedness, where the definition of adjacency depends on the used move operator. A high correlation between neighbored solutions indicates very similar fitness values and therefore reveals a smooth landscape. Another established indicator is the autocorrelation measured among the time series steps of random walks. Garrett proposes the autocorrelation also as an indicator for ruggedness in the case of MOPs. Especially the autocorrelation between Pareto optimal solutions provides information about the opportunity to find further Pareto optimal solutions by a guided local search starting from a local optima, in contrast to the performance of a random restart for the search of further optimal solutions.

Some work has been done regarding multi-objective landscape analysis for combinatorial optimization problems, especially for the multi-objective QAP. Garrett provides a framework for multi-objective fitness landscape analysis in his thesis on the basis of multi-objective fitness distance correlation, a directed and undirected random walk analysis from one Pareto optimal solution to another one as well as a random walk and basin analysis. Furthermore, Garrett defined the multi-objective generalized assignment problem and compared its landscape structure to those of the multi-objective QAP. Knowles and Corne [96] presented a search algorithm to solve the multi-objective QAP that is designed with regard to the multi-objective landscape structure. A number of landscape measures and techniques are proposed to explore the landscape of the multi-objective QAP. Generally, the landscape is analyzed by a rough objective correlation.

Garrett and Dasgupta [71] presented the plateau connection structure analysis, where the search space is divided into a set of equivalent classes according to the Pareto dominance relation. The probabilities of a specific neighborhood structure like moving from one class to another one is empirically investigated. This concept has been examined on a number of benchmark multi-objective combinatorial optimization.
Aguirre and Tanaka [2] analyzed the properties of MNK-landscapes by an enumeration on small landscapes: the number of fronts, the number of solutions on each front, the probability to pass from one front to another and the hypervolume of the Pareto optimal set.
Verel et al. [165] defined a further type of multi-objective fitness landscape analysis on the basis of detecting search space properties of defined set levels. The contribution of this work is the definition of the multi-objective fitness landscapes based on a search space divided into solution sets. Furthermore, a neighborhood search operator is defined for the exploration between the solution sets, and an indicator-based fitness function is proposed. Empirical studies have been performed on multi-objectives NK-landscapes.
In a further work, Verel et al. [164] proposed a multi-objective combinatorial search space analysis with the focus on the correlation between the objective functions and with the aim of providing guidelines for the design of a multi-objective local search algorithm based on the main fitness landscape properties. The multi-objective NK-landscapes have been extended to take account of the objective correlation. More precisely, the co-influence of the problem dimension, the degree of non-linearity, the number of objectives and the objective correlation on the structure of the Pareto optimal set in terms of cardinality, number of supported solutions as well as the number of Pareto local optima are investigated.

3.3 Concepts of Multi-objective Molecular Landscapes Analysis

The aim of multi-objective landscape analysis is the design of a metaheuristic algorithm with optimized performance. The structure of the search space gives some information about an adequate search pattern of the metaheuristic: How - and more general - is the metaheuristic able to outperform a random search in discovering of optimal solutions by guiding the search from a previous optimal solutions to other ones? By answering these questions, the evaluation

of the landscape analysis provides a hint for a suitable explorative behavior of the search algorithm. In general, local properties strongly influence the effectiveness of local search, whereas global properties strongly influence the effectiveness of a metaheuristic [111].
MOML analysis is performed with the aim of providing an analysis concept that characterizes the multi-dimensional molecular landscapes regarding the important landscape properties modality or optima density, correlation or linear relationship between the molecular fitness functions, ruggedness and neutrality in form of plateaus. Suitable analysis techniques are intuitive with regard to the landscape composition according to Definition 6 and the investigated properties. Furthermore, these techniques are simple to calculate and most important - independent of the optimization problem dimension. Inspired by the work of Emmerich and Lee, the analysis techniques are based on random walks with one-point mutation as move operator to investigate the neighbored molecular landscape. The one-point mutation is used as move operator for an insight into the mutation potential of a MOEA and to avoid highly differing consecutive configurations, which are potentially produced by a recombination operator. Small changes in the configurations provide information about the effectiveness of the local search of a MOEA, whereas large changes in the configurations - potentially provided by a higher mutation probability or the recombination operator - provide an insight into the effectiveness of the global search pattern of a MOEA. Another argument for the use of the one-point mutation is the verification of the similarity hypothesis for molecules: The similarity of the molecule structure is often related to the similarity of the molecule properties. In general, the start solution of the random walk is initialized randomly. The objectives of the multi-objective fitness landscapes are real-valued vectors of the length k according to the number of objective functions.

Modality. Modality (i.e., the investigation of optima density) is examined based on measurements of the random walk part consisting only of non-dominated solutions - or said differently - the individuals of the first front. According to Garrett, these individuals are the global optima. The modality requests information about the number of non-dominated solutions, a potential clustering of those or otherwise a large distribution over the MOML. For this purpose, the individuals of the random walk are ranked into fronts. For an Optima Distribution Analysis (ODA), the average Euclidean distance d_{ODA} between all possible

combinations of non-dominated fitness values $\vec{x}_i$ (solutions) is determined.

$$d_{ODA} = \frac{1}{K} \sum_{i,j} d_{ij} \text{ with } d_{ij} = |\vec{x}_i - \vec{x}_j| \text{ for } i, j = 1, ..., M \text{ and } i < j, \quad (3.13)$$

where M is the number of fitness vectors in the first front and $K = \binom{M}{2}$ the number of all possible combinations of differences d_{ij}. In the case that the range of the objective functions are very differently to one another, the use of the normalized Euclidean distance is advisable. The value of d_{ODA} is a measure for the central tendency of non-dominated solution diversity and d_{ODA} is regarded as the mean distance between all non-dominated solutions. In the statistical sense, a mean has its limitation in the case of extremal boundary values as the calculated mean value is skewed in the case of extremal values in the data set. To overcome this problem, the diversity of the non-dominated solutions is quantified by the average deviation of all distances d_{ij}:

$$d_{MAD} = \frac{1}{K} \sum_{i,j} |d_{ij} - \bar{d}| \text{ with } i, j = 1, ..., M \text{ and } i < j. \quad (3.14)$$

with $\bar{d} = d_{ODA}$. This diversity measure is also used to evaluate MOEA performance and is further discussed in section 5.2.4. The higher the diversity value, the wider is the spread of the non-dominated solutions over the search space.

Another indicator for the distribution of non-dominated solutions is the measurement of the beeline between two consecutive non-dominated solutions along the random walk path (Figure 3.1). Therefore, the magnitude of the beeline between two fitness vectors $\vec{x}_{i+1}$ and $\vec{x}_i$ (symbolizing the non-dominated fitness vectors ordered according to their position in the random walk) is determined and set in relation to $\bar{c} = \frac{1}{N-1} \sum_{i=1}^{N-1} |\vec{y}_{i+1} - \vec{y}_i|$, the average Euclidean distance between two consecutive fitness vectors $\vec{y}_{i+1}$ and $\vec{y}_i$ of the random walk with N as the number of random walk steps to classify the distribution tendency:

$$b_i = \frac{|\vec{x}_{i+1} - \vec{x}_i|}{\bar{c}} \text{ with } i, j = 1, ..., M - 1. \quad (3.15)$$

A low number of relational beelines b_i indicates that the corresponding distance between two consecutive non-dominated solutions is relatively small compared to the average distance between the consecutive distances of the random walk.

Correlation. Correlation coefficients are commonly known as a measurement for the strength of (linear) association between variables [147]. Therefore, the correlation is usable as a measure for the relationship between two solutions

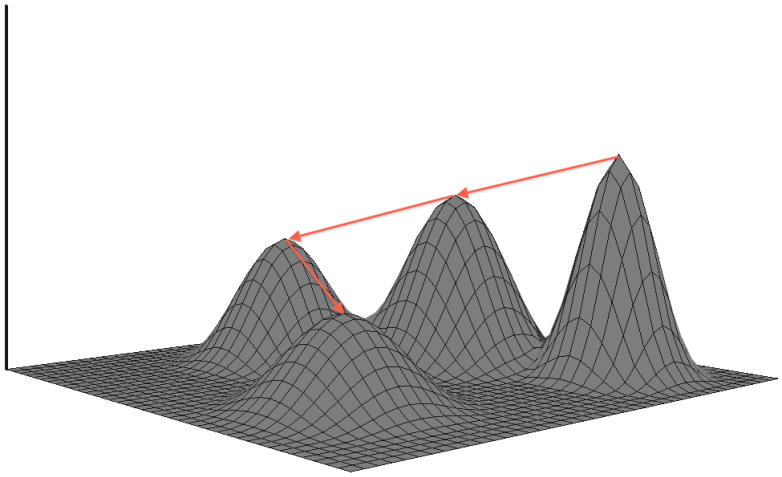

Fig. 3.1: Beelines between the consecutive non-dominated solutions depicted on an idealized landscape.

in a landscape. A correlation analysis of the single fitness functions provides some information about the correlation tendency of the corresponding fitness values. In the case of MOMLs, the correlation between the single molecular fitness functions is of great interest, because the high correlation between two time series of different fitness functions theoretically reduces the optimization problem dimension and therefore the problem difficulty. The correlation matrix is a suitable analysis technique for this purpose:

$$M_{corr} = \begin{pmatrix} 1 & corr(f_1, f_2) & ... & corr(f_1, f_k) \\ corr(f_2, f_1) & 1 & ... & corr(f_2, f_k) \\ \vdots & \vdots & \ddots & \vdots \\ corr(f_k, f_1) & corr(f_k, f_2) & ... & 1 \end{pmatrix}, \tag{3.16}$$

where M_{corr} is symmetrical and consists of the Pearson correlation coefficients of the fitness function f_i and f_j:

$$corr(f_i, f_j) = \frac{\sum_{i=0}^{n}(f_i - \bar{f}) \cdot (f_j - \bar{f})}{\sigma_{f_i} \cdot \sigma_{f_j}} \tag{3.17}$$

In this context, the correlation coefficients lie in a range of $[-1; 1]$, where a negative value symbolize a potential anti-proportional linear relationship and a positive value a possible proportional linear relationship. Furthermore, no or at least a low correlation is given by $|corr(x, y)| < 0.3$. A moderate correlation is given by $0.3 \leq |corr(x, y)| \leq 0.8$ and $|corr(x, y)| > 0.8$ indicates a high linear correlation.

Ruggedness. The ruggedness refers to the relationship between each solution and its neighbors. A landscape is said to be rugged if it reveals high varying fitness values: the greater the fitness differences, the more rugged is the landscape. From this point of view, the analysis technique for MOML ruggedness is based on the difference vectors determined between each two consecutive fitness vectors of a random walk. A measure for the variation of the fitness vector values is the magnitude of the absolute value calculated of the difference vectors. The absolute value of the difference vectors provides an insight in the magnitude of differences between the single molecular fitness functions. A closer consideration of the absolute values as a measure for fitness difference - in the case of MOML - leads to the insight that this value does not take account of the fitness variation of the different single molecular fitness functions in the sense that potentially only a few of the fitness functions are responsible for a high absolute value. Furthermore, another view on the absolute value reveals that it is no indicator for the direction of the single molecular function moving and therefore no indicator for the increase, decrease or stagnation of the different fitness functions. These considerations lead in conclusion to a definition of ruggedness for real-valued multi-objective fitness landscapes: A real-valued multi-objective fitness landscape is regarded as rugged if the single fitness functions are moving differently with high fitness differences. As a consequence, this landscape is regarded as smooth if all fitness functions are moving equally or only a very few of these functions are directed differently and with small fitness differences.
The information about the single fitness function directions are provided by the difference vectors between the consecutive fitness vectors. A suitable indicator for the direction of the difference vectors is the angle between the difference vectors as - in general - an angle between vectors is an indicator for similarity [151]:

$$similarity(\vec{x}, \vec{y}) = cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|} \tag{3.18}$$

This is the key connection between linear algebra and probability theory: The vectors $\vec{x}$ and $\vec{y}$ are centered by subtracting the means $\vec{x}\,'$, $\vec{y}\,'$ from these vectors:

$$\vec{u} = \vec{x} - \vec{x}\,' \text{ and } \vec{v} = \vec{y} - \vec{y}\,'. \tag{3.19}$$

Then, it holds

$$cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|} = \frac{\sum_{i=1}^{k}(x_i - x_i\,') \cdot (y_i - y_i\,')}{\sqrt{\sum_{i=1}^{k}(x_i - x_i\,')^2} \cdot \sqrt{\sum_{i=1}^{k}(y_i - y_i')^2}} = corr(\vec{u}, \vec{v}), \tag{3.20}$$

For the angle between two consecutive difference vectors, which provide information about the relative position of three consecutive fitness vectors and therefore of the fitness variance direction of a solution and its neighbors along the random walk, the following geometrical interpretation is stated: an angle of 0° refers to two vectors pointing in the same direction. This implies that the single fitness values of the consecutive random walk steps, which define these two difference vectors, are all positioned in the same direction. Otherwise, an angle of more than 90° indicates a moving of a majority of the single fitness functions in different directions. In the case of the stagnation of all objective function values, the different vector is the zero vector and the angle is not defined. Then, the angle is set to 0°.
Hence, to gain an insight into the potential ruggedness and structure of the MOML, the angle between each two consecutive difference vectors are calculated. Furthermore, the random walk path length of the difference vectors enclosing a particular angle $\measuredangle(x_{i+1}, x_i) = a$ with $a \in [0; 180]$ is determined to gain information about the magnitude of fitness differences. The summarized length of this path parts allows no statistically interpretation, since this value depends on the subspace dimension of the search space relative to the random walk steps. Therefore, this path length is set in relation to the total number of random walk steps.

$$p_{length,a} = \frac{\sum\limits_{i,\ \measuredangle(x_{i+1},x_i)=a} |x_{i+1}| + |x_i|}{N-1} \text{ with } a \in [0; 180]. \tag{3.21}$$

Plateaus. Plateaus are another important structure of a landscape. The number and size of plateaus are investigated by neutrality measures [55]. In MOMLs, plateaus are characterized in two different aspects: Firstly, plateaus are characterized according to the stagnation of all objective functions values over several steps of the time series and secondly - from a more global view - according to the number of consecutive time series steps in the same Pareto front. The plateau characterization in the sense of objective function stagnation is determined via:

$$|x_{i+1} - x_i| \leq 1 \text{ for } i = 1, ..., N-1. \tag{3.22}$$

These proposed analysis techniques are applied on random walks of different lengths. The mutation of the same amino acid is excluded to avoid a stagnation of the random walk. For statistical reasons, these random walks are repeated at least 30 times. The methods of the statistical evaluation and representation are described in section 5.3.

3.4 Molecular Fitness Functions

A MOP is composed of several fitness functions also denoted as objective functions. Each fitness function returns a value that reflects the quality of a particular solution in solving the MOP. In general, these fitness functions have to be generated and most efficiently implemented to reduce the computational complexity, since a high number of fitness function evaluations have to be performed in a MOEA run. The number of fitness function evaluations depends on the number of objective functions and the number of performed generations.
The benchmark problems used in this thesis are composed of four different objective functions describing physiochemical properties of peptides. These functions have been selected under different aspects: The functions have to compose a benchmark problem that is as generic as possible, firstly in the sense that the predicted properties allow conclusions on a range of peptide features. Secondly, there exist a range of common approximate calculation methods for physiochemical peptide properties, which are calculated by the specific descriptor values of the amino acids contained in a peptide sequence. Three of the four fitness functions are based on such an approximate calculation. Furthermore, the functions are associated to the three structural levels of peptides: Two physiochemical objective functions refer to the primary structure[8] of a peptide. One objective function gives information about the secondary structure[9] and the last one gives a clue about possible early tertiary structure[10] disruption or an inadequate folding. These four objective functions act comparatively as they reflect the similarity of a particular peptide or solution to a pre-defined reference peptide. This is implemented by the difference between the fitness function value of a particular peptide and the reference peptide:

$$f(CandidatePeptide) := |f(CandidatePeptide) - f(ReferencePeptide)| \quad (3.23)$$

Therefore, the four objective functions have to be minimized and the benchmark problem is a minimization problem according to eq. (1.1). The fitness function values $f(CandidatePeptide)$ are always positive realized by the use of the absolute value on the difference.

[8] The primary structure is the linear sequence of the involved amino acids. The primary structure is listed starting from the free amino acid group.

[9] The secondary structure describes regularities of the local structure referred to a few amino acids. Therefore, the secondary structure is determined to a large extend by the primary structure.

[10] The tertiary structure describes the spatial organization of a peptide. Elements of the secondary structure are grouped together as domains. The spatial organization of a peptide is responsible for its effect.

Three of the objective functions are implemented via the open source project BioJava [123]. BioJava is a Java tool that provides different tools to compute physiochemical properties as well as a module for sequence alignment for peptides and proteins composed of the 20 canonical amino acids. In the following, the fitness functions are motivated and described.

3.4.1 Needleman-Wunsch Algorithm

The motivation for the Needleman-Wunsch Algorithm (NMW) [115] is the detection of similarities between peptides or proteins with regard to biochemical functionality and structure. This similarity is qualitatively recognized by a global sequence alignment. NMW is implemented by the BioJava library. The main goal of NMW is to find the optimal global alignment of a candidate solution peptide to a pre-defined reference peptide. The optimal global alignment is detected by maximizing the number of amino acids matches and minimizing the number of gaps to align the candidate solution peptide and the reference peptide. NMW makes use of dynamic programming and allows the use of optimal scoring models.
Dynamic programming ensures to find the optimal alignment in a quantitative way by assigning scores for matches, mismatches, gaps and stores them into a scoring matrix. NMW allows the use of different scoring models in the way that different values are assigned for matches or mismatches. Special scoring matrices are pre-defined in the case of amino acid substitutions. Here, the BLOcks SUbstitution Matrix (BLOSUM) developed by Henikoff and Henikoff [81] is used in form of the percentage identity 100 (BLOSUM100)[11]. After filling the matrix, the last step is the tracing back through the matrix elements for the best alignment. The highest scoring path through the matrix gives the best alignment. The computational complexity of NMW is $O(M \cdot N)$, where M and N are the sequence lengths of the two peptides or proteins.

3.4.2 Molecular Weight

The design of a new drug candidate for oral bioavailability requires a number of specific molecular properties. One of these important properties is the Molecular Weight (MW) [162] to ensure oral bioavailability. The BioJava library also provides the objective function calculating the molecular weight of

[11] BLOSUM100 is derived from sequence alignment with 100% identity.

the candidate solution peptides. MW of a peptide sequence of the length l is determined in a most simple way, by the sum of the mass of each amino acid (a_i) plus a water molecule:

$$\sum_{i=1}^{l} mass(a_i) + 17.0073(OH) + 1.0079(H), \tag{3.24}$$

where O (oxygen) and H (hydrogen) are the elements of the periodic system. The computational complexity for the determination of the MW of a peptide or protein with the length l is $O(l + 1)$.

3.4.3 Average Hydrophilicity

Hydrophilicity is one of the most important physiochemical parameters in the area of drug design [148]. Hydrophilicity reflects the tendency of a peptide to be dissolved in aqueous solutions. One of the main goals in peptide-based drug design is the optimization of the hydrophilic-hydrophobic balance as drug characteristic. On one side, hydrophobic peptides govern the biological processes like transport through body fluids, metabolism and folding. The drawback of hydrophobic drugs is an inhibition of biological systems compared to hydrophilic ones and they are slower eliminated by a biological system. On the other side, the hydrophilic character of peptide-based drugs is essential for the ability to cross cell membranes. [80]
The hydrophilic character of a candidate solution peptide is determined by the method and hydrophilicity scales of Hopp and Woods [86]. This method is evolved to identify the potential antigenic sites in proteins. Hopp and Woods determine hydrophilic parts of a peptide or protein by sliding a window of fixed size over the sequence. The hydrophilic scales of the amino acids within the window are averaged. The window size refers to the number of amino acids examined to identify the hydrophilic character of this peptide or protein site. The advisable window size is six [85].
In this thesis, a hydrophilicity value is assigned to each candidate solution peptide with a window size equal to the peptide length l using the scales of Hopp and Woods for each amino acid a_i:

$$\frac{1}{l} \cdot (\sum_{i=1}^{l} hydro(a_i)). \tag{3.25}$$

The computational complexity for the determination of the average hydrophilicity of a peptide or protein with the length l is $O(l)$.

3.4.4 Instability Index

Peptides are used as highly specific and effective therapeutic agents, but their use is potentially restricted by their instability. They are usually evolved for targets inside cells [154].
The peptide conformation is responsible for the disruption of the tertiary structure or an unfavorable folding. Guruprasad et al. proposed a method to predict the instability characteristic of a peptide by using a sliding window of the length of two amino acids to analyze the primary structure of a peptide sequence with the aim of predicting the potential intracellular instability of a peptide [78]. More precisely, the Dipeptide Instability Weight Values (DIWV) of each two consecutive amino acids in the peptide sequence are summarized and the final sum is normalized by the peptide length l:

$$Instability\ Index = \frac{10}{l} \sum_{i=1}^{l} DIWV(x_i, x_{i+1}) \tag{3.26}$$

DIWV are provided by the GRP-Matrix. This matrix has been constituted by statistical analysis of the primary structure on different sets of unstable and stable proteins and contains the condition-based instability weight values for the 400 possible dipeptide compositions [78].
The computational complexity for the determination of the average hydrophilicity of a peptide or protein with the length l is $O(l + 1)$.

3.5 Analysis of Single Molecular Fitness Functions

A first insight into the four molecular landscapes - NMW, MW, Average Hydrophilicity (Hydro) and Instability Index (InstInd) - is exemplarily illustrated by six random walks over each fitness function landscape of a length of 100 in Figure 3.2 to Figure 3.5. All four molecular fitness functions provide large variations of the fitness values over the 100 random walk steps and therefore indicate rugged landscapes. From the global point of view, NMW reveals some plateaus over two to five random walk steps as well as some areas with smaller fitness value differences (Figure 3.2). The InstInd function also reveals some plateaus, but to a lesser extent and on average over a lower number of random walk steps (Figure 3.5). The fitness values of the MW function are scaled by a factor of 10 and achieve large jumps of the fitness values as well as some areas

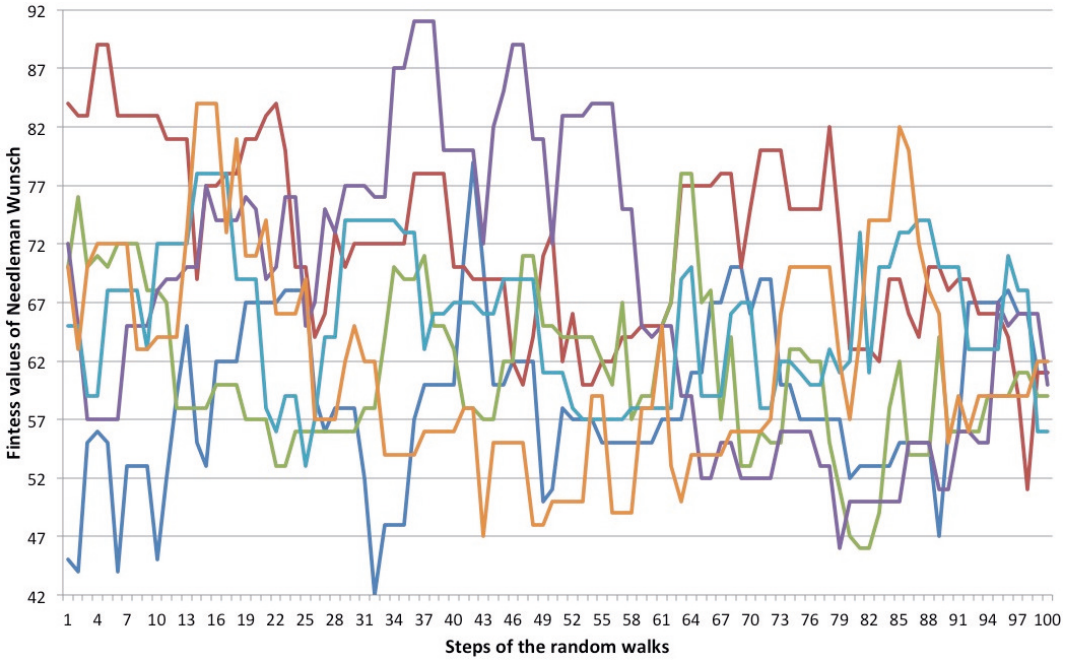

Fig. 3.2: Needleman-Wunsch fitness function values of six random walks.

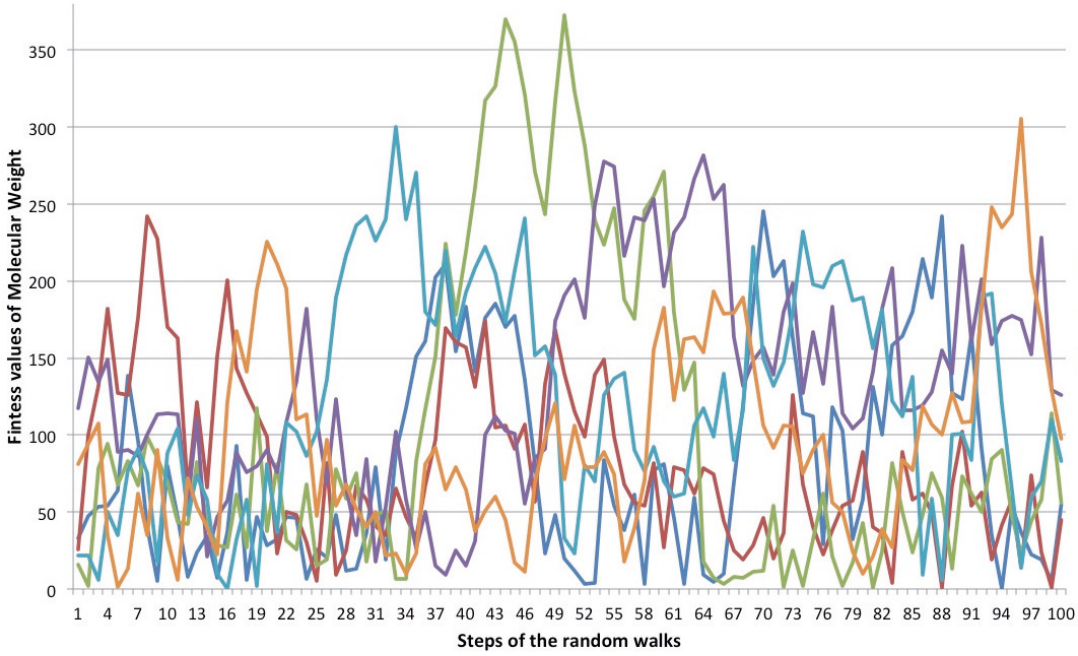

Fig. 3.3: Molecular Weight fitness function values of six random walks.

with oscillating parts with a low frequency (Figure 3.3). The Hydro fitness function appears similar to MW regarding the jumps and the oscillating parts (Figure 3.4). Otherwise, it also reveals some isolated flat areas or plateaus.

The rugged properties of the four molecular landscapes are further quantified via the autocorrelation function $p(s)$ (eq. (3.10)) as introduced by Emmerich et

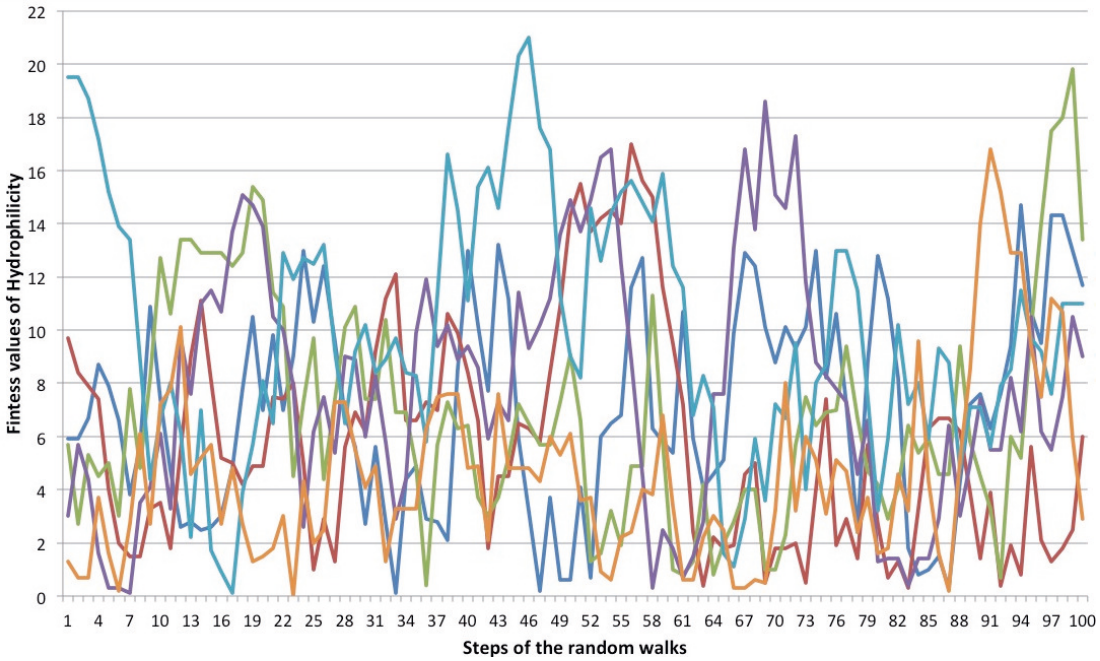

Fig. 3.4: Average Hydrophilicity fitness function values of six random walks.

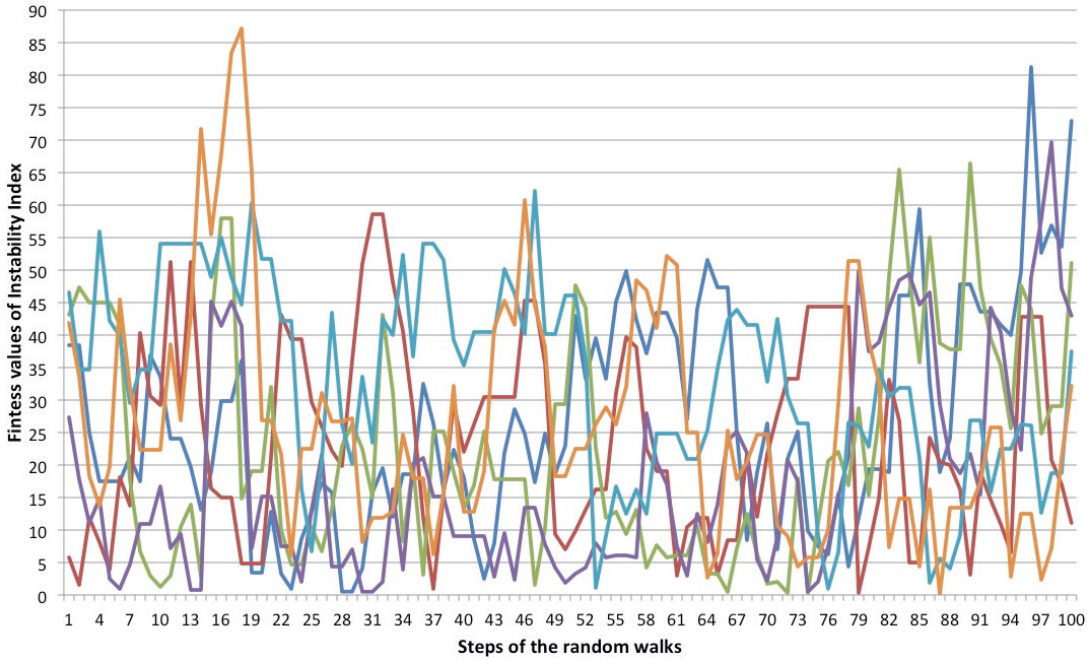

Fig. 3.5: Instability Index fitness function values of six random walks.

al. [55]. The time series to determine the autocorrelation functions are composed of 30 random walks of a length of 100. The start configuration is randomly determined. In general, all autocorrelation functions decrease from $p(0)$ on and reveal a more or less strong oscillating behavior mainly in the range of -0.3 and $+0.3$ apart from some outliers (Figure 3.6). This range statistically indicates a

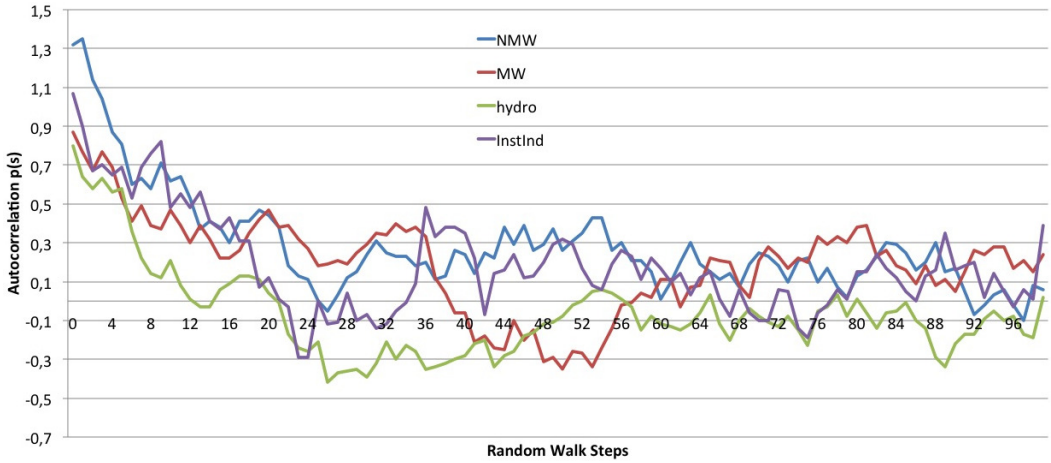

Fig. 3.6: Autocorrelation p(s) for NMW, MW, Hydro and InstInd over 100 random walk steps.

weak correlation. The outliers of the autocorrelation values after $p(13)$ are up to 0.5 and down to -0.4, which indicates only a moderate correlation. Therefore, the autocorrelation functions of all molecular time series reveal mostly weak correlation values and as a consequence, the four molecular landscapes are highly rugged.

In general, the self-correlation coefficient $p(0)$ is approximately 1 for the four fitness function. This indicates that the mean squared deviation of the starting points is approximately equal to the variance of all configurations of the performed random walks (eq. (3.10)). The time series on InstInd have the lowest self-correlation coefficient with $p(0) = 0.87$ and the time series of NMW have the highest value with $p(0) = 1.32$ compared to the other functions. The autocorrelation values of Hydro reveal the strongest decrease, which indicates the highest ruggedness. Moreover, the highest ruggedness is indicated by the Hydro autocorrelation values, which are weakly oscillating around -0.1 from $p(45)$ on. The MW autocorrelation values provide the slowest decrease after $p(0)$ as well as the highest number of outliers from the weak correlation range. The autocorrelation values of NMW increases only for $p(1)$ and the times series of NMW provide the lowest number of negative autocorrelation values.

A further landscape analysis of the four single molecular landscapes is performed by a quantitative MOML analysis. The dimension of the MOML depends on the correlation or the relationship between the participating molecular fitness functions. Therefore, the correlation matrix (3.16) has been determined for the four molecular functions (NMW (f_1), MW (f_2), Hydro (f_3) and InstInd (f_4)). The 30 time series of all molecular functions over 100 random walk steps serve as data set for the determination of the correlation matrix (Figure (3.7)): The matrix entries reveal only low correlations between the time series of each

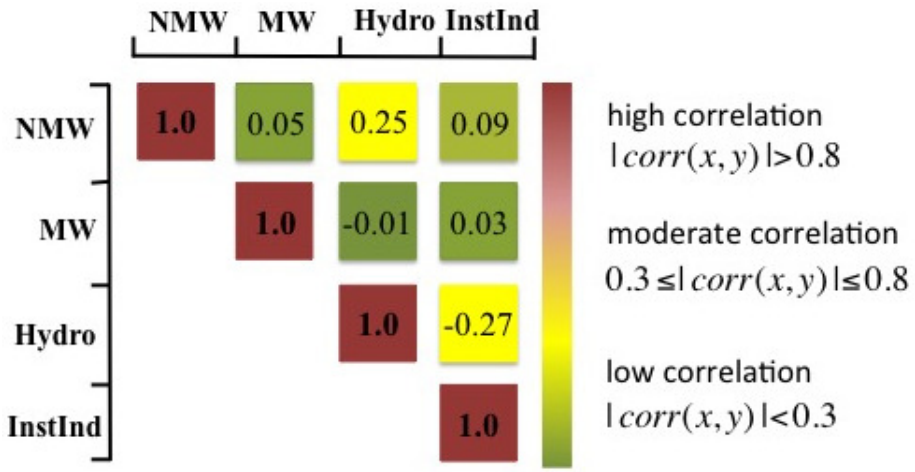

Fig. 3.7: Correlation plot of the molecular fitness functions NMW (f_1), MW (f_2), Hydro (f_3) and InstInd (f_4).

two molecular fitness functions: The highest correlation values beneath the diagonal of the matrix are between NMW and MW (eq. (3.17): $corr(f_1, f_3) = 0.25$) as well as between InstInd and Hydro (eq. (3.17): $corr(f_3, f_4) = -0.27$). As a consequence, the dimension of a MOML constituted of these four molecular functions is equal to the number of participating objective functions.
In the following, the 3D-MOML is first analyzed followed by the 4D-MOML on the basis of the presented MOML analysis concept.

3.6 3D Molecular Landscapes

The 3D-MOML is constituted of the three molecular functions NMW, MW and Hydro. The 3D-MOML is characterized by modality, ruggedness and plateaus according to the proposed concept in Section 3.3. The modality of the 3D-MOML is first investigated on the basis of the number of non-dominated solutions identified in a random walk of a length of 100 and 500 respectively. For statistical reasons, 50 random walks of each length have been performed and boxplots have been created to depict the number of non-dominated solutions (Figure 3.8) and the number of detected fronts in the random walks (Figure 3.9). The left boxplot of Figure 3.8 reveals that the middle 50% of the non-dominated solutions or the solutions in the optimal front are in the range of 10% to 17% of the random walk length represented by the inter-quartile range of the boxplot. The median divides the box equally with a value of 13. An increase of the random walk length (right boxplot) results in an increase

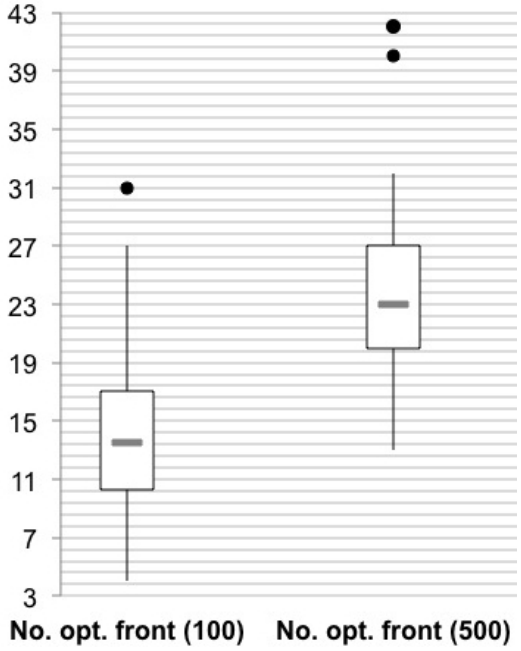

Fig. 3.8: Number of non-dominated solutions in time series of length 100 and 500.

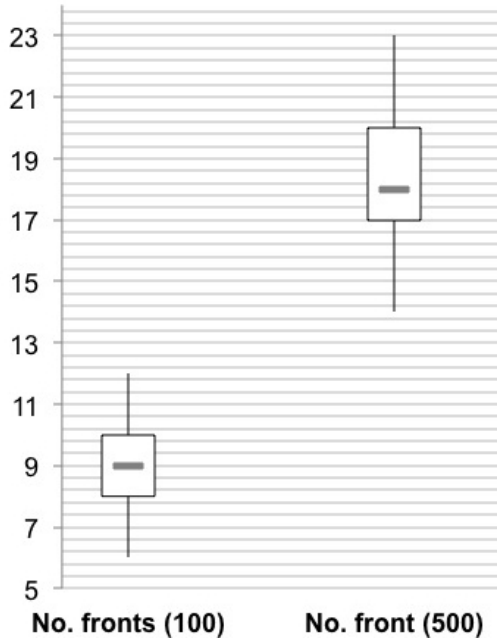

Fig. 3.9: Number of detected fronts in time series of length 100 and 500.

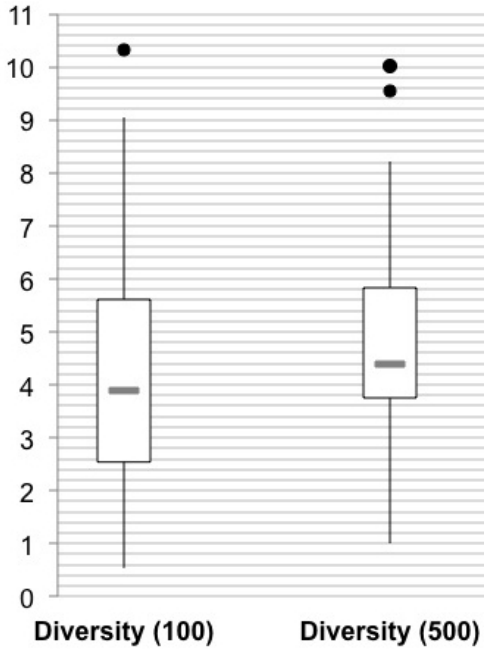

Fig. 3.10: Diversity (d_{MAD}, eq. (3.14)) measured of the non-dominated solutions obtained by time series of a length of 100 and 500.

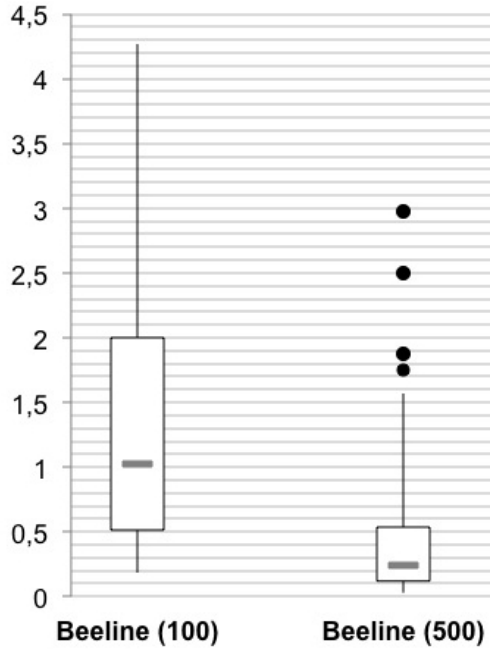

Fig. 3.11: Beeline distance of two consecutive non-dominated solutions relational to the average distance between all consecutive solutions of the time series (eq. (3.15)).

of the solution number in the optimal front by approximately 83.9%[12] regar-

[12]This value is the geometric mean applied on the percentage increase of the five boxplot values. The geometric mean characterizes the global tendency of the averaged percentage

ding to the results of the random walks with a length of 100. The black dots in the figure represent the outliers indicating some remarkably higher results. Concluding, the investigation of larger times series of the 3D-MOML achieves a larger solution number in the optimal front, but this increase is of a lower percentage level than the increase of the random walk length. The reason for this observation is the increase of the detected number of fronts from the random walks of the length 500 compared to those of the length 100 (Figure 3.9). The left boxplot of Figure 3.9 reveals that the middle 50% of the detected fronts are in a range of 8 to 10 fronts represented by the inter-quartile range. The median divides the box equally with a value of 9. An increase of the random walk length results in an increase of the detected number of fronts by approximately 104% regarding the results of the time series of a length of 100. The front numbers 17 and 20 determine the inter-quartile range of the right boxplot. Concluding, the increase of the solution number in the optimal front is of a lower level, since the number of fronts increases significantly by an increase of the time series length.

Figure 3.10 depicts two boxplots for the diversity of the non-dominated solutions, once more calculated of 50 random walks of a length of 100 (left boxplot) and 500 (right boxplot) respectively. The left boxplot reveals a large spread of the inter-quartile range and therefore indicates that there exists a wide range of diversity abilities within the non-dominated solutions of the time series. The increase of the time series length results in an average increase of 24.4%.

The relational beeline between each consecutive non-dominated solution has been determined for 30 random walks of a length of 100 and 500 respectively. The left boxplot in Figure 3.11 depicts the averaged five boxplot values of (eq. 3.15) of the 30 random walks. The spread of the inter-quartile range indicates that some of the non-dominated solutions are more clustered in the landscape (lower quartile of $0,5$ indicates that the distance between the consecutive non-dominated solutions is half of the average distance between all consecutive solutions of the time series) and some are positioned in a wide distance (25% of the relational beeline values are above 2.0, which indicates that the distance between the consecutive non-dominated solutions is more than twice of the average distance between all consecutive solutions time series). The right boxplot in Figure 3.11 represents the spread of the relational beelines between the non-dominated solutions in the random walks of the length 500. The comparison of both boxplots in this figure reveals that an increase of the random walk length results in a significant decrease of the relational beeli-

increase/decrease of the boxplot.

ne lengths between the non-dominated solutions, whereas the right boxplot of Figure 3.11 reveals some outliers indicating some high distances between the non-dominated solutions of the random walks with the length 500. These results allow the conclusion that the non-dominated solutions are mainly clustered in the MOML.

The relational beeline and the number of non-dominated solutions - as already

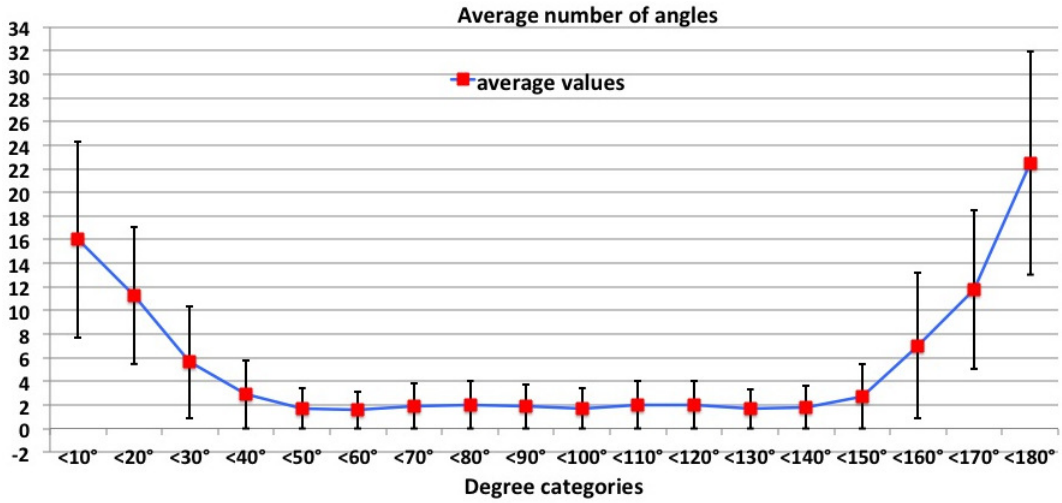

Fig. 3.12: Average number of angles between two consecutive difference vectors categorized in degree intervals of the length 10.

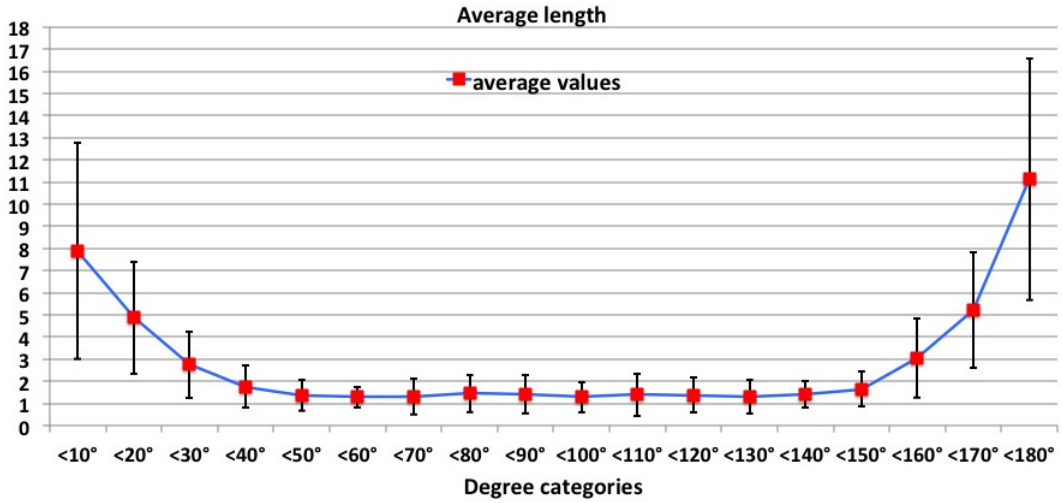

Fig. 3.13: Average length of each two consecutive difference vectors enclosing a particular angle (eq. (3.15)). The angles are categorized in degree intervals of the length 10.

proposed above - provide a rough inside into the ruggedness of the 3D-MOML. For a quantified insight in this ruggedness, the difference vectors between all consecutive solutions of the time series with a length of 100 are determined

and the angles between each two consecutive difference vectors are calculated (eq. (3.20)) to gain an insight into the movement characteristics of the single objective functions. The angle between two consecutive difference vectors gives information about the similar movement behavior of the single objective function over three consecutive solutions of the time series.
Figure 3.12 depicts the average number of angles - categorized in intervals of ten degree on the x-axis - over the 30 random walks of the length 100. The depicted upper and lower boundaries mark the 95%-significance interval. The highest number of angles is detected in the interval of $[170°; 180°)$. This indicates that the difference vectors are oppositely directed and the single objective functions are increasing, decreasing or stagnating over three steps of the time series in very different manners. Exemplary spoken: One objective function increases from a time series step to the next one and decreases afterwards. The second function is moving exactly the other way around and the third function is stagnating from the first to the second solution and increasing or decreasing afterwards.
This reveals that the landscape is very rugged along a large number of random walk steps. The second highest number of angles is in the interval of $[0°; 10°)$. This indicates that the difference vectors are similarly directed and the single objective functions are increasing, decreasing or stagnating in a similar manner. Exemplary spoken: One of the objective functions is stagnating over three time series steps and the other two functions are increasing or decreasing over these three steps. The number of angles in the interval of $[40°; 150°)$ are almost stable. The larger the angle, the larger the number of objective functions revealing oscillating moving behavior in different manner over three steps of the random walks.
A similar pattern is achieved by calculating the average path length with a particular bending (Figure 3.13) provided by the difference vectors, which enclose specific angles categorized once more in intervals of 10° (eq. (3.21)). The highest length is achieved in the interval $[170°; 180°)$ indicating large differences between the single molecular function values with mainly oscillating behavior. The second highest length is achieved in the interval $[20°; 30°)$ indicating large differences between the solutions of the time series, which are mostly positioned in the same direction. The length of the difference vectors enclosing angles in the interval $[40°; 150°)$ are small and reveal therefore only slight changes of the single objective function values.

Plateaus are a further structural property that provides some information about clustered similar qualified solutions. Firstly, plateaus are identified in MOMLs by consecutive equal or nearly equal fitness values for each molecular

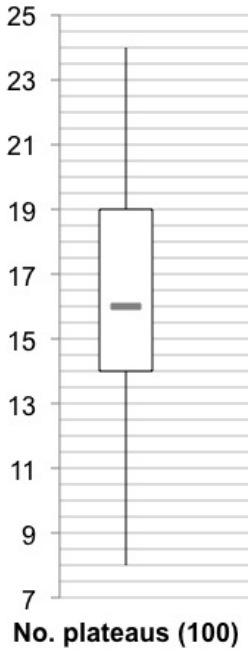

Fig. 3.14: Average number of plateaus characterized by consecutive time series steps in the same Pareto front in time series of the length 100.

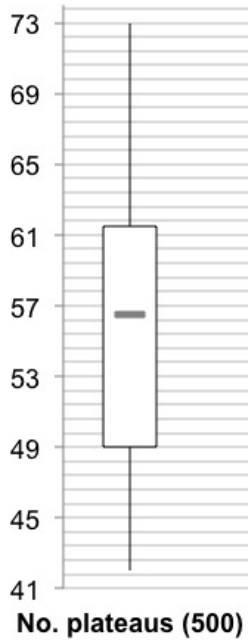

Fig. 3.15: Average number of plateaus characterized by consecutive time series steps in the same Pareto front in time series of the length 500.

function (see eq. (3.22)). In 30 random walks of length 100, 20 plateaus have been identified totally: Two plateaus, each consisting of two consecutive equal fitness values, have been identified in five random walks. A plateau of three consecutive equal fitness values has been found in one random walk and the remaining 9 plateaus have been identified in different random walks, each consisting of two consecutive equal fitness values.

Secondly, plateaus are characterized more globally by consecutive time series steps in the same Pareto front. The Figures 3.14 and 3.15 depict the number of consecutive solutions assigned to the same Pareto front within time series steps of the length 100 (Figure 3.14) and 500 (Figure 3.15) respectively. In the time series of the length 100, 50% of the plateaus numbers are in the range of 14 to 19 with a median of 16 plateaus. An increase of the time series length results in an increase of the plateaus by approximately 264%. Thus, the increase of the plateau number is significantly lower than the increase of the time series length. This is once more a consequence of the high front diversity within larger time series (see Figure 3.9). $14,5\%$ of the plateaus detected in the time series of the length 100 are first front plateaus. In the time series of the length 500, only 7% of the plateaus are first front plateaus. This is once more a consequence of the larger front diversity. The magnitude of the plateaus is defined by the number of consecutive solutions in the same Pareto front. The

average plateau size is decreased from 2.31 to 2.18 by the increase of the time series length with a standard deviation of 0.033 and 0.012 respectively. The average plateaus size of the first front plateaus is on average larger with 2.7 and 2.3 (standard deviation of 0.13 and 0.11 respectively) in the time series of length 100 and 500. The plateaus are distributed unevenly along the random walks: Some plateaus are only separated by one time series step, others by a wide range of steps.

3.7 4D Molecular Landscapes

The 4D-MOML is constituted by inclusion of the molecular function InstInd to the 3D-MOML. More precisely, the 4D-MOML consists of the molecular functions NMW, MW, Hydro and InstInd. The 4D-MOML is characterized by modality, ruggedness and plateaus and the results of these structural properties are discussed in comparison to the landscape properties of the 3D-MOML. Furthermore, a direct comparison of the 3D- and 4D-MOML based on the achieved non-dominated solutions identified in a random walk is visualized.

The modality of the 4D-MOML is quantified by the number of non-dominated

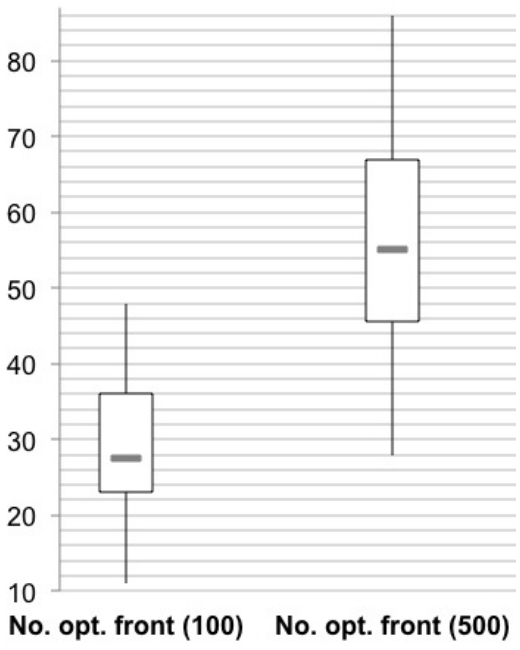

Fig. 3.16: Number of non-dominated solutions in time series of a length of 100 and 500.

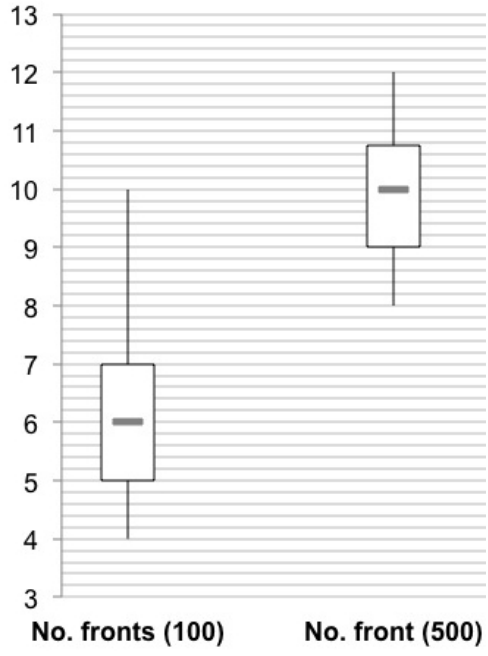

Fig. 3.17: Number of detected fronts within the time series of a length of 100 and 500.

solutions detected in random walks of the length 100 and 500 respectively (Fi-

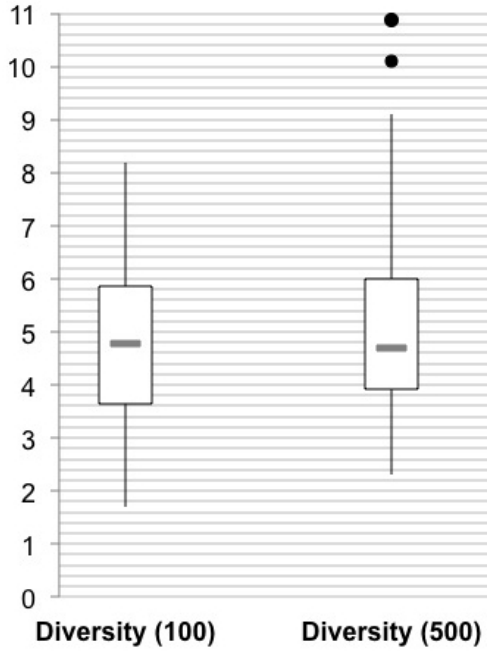

Fig. 3.18: Diversity (d_{MAD}, eq. (3.14)) measured of the non-dominated solutions obtained by random walks with a length of 100 and 500.

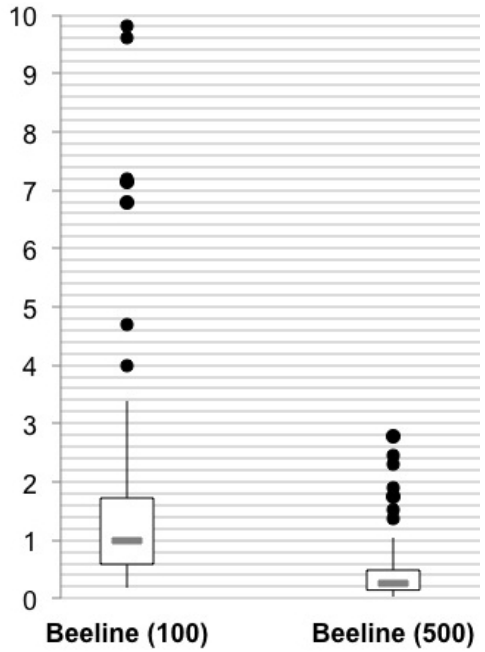

Fig. 3.19: Beeline distance of two consecutive non-dominated solutions relational to the average distance between all consecutive solutions of the time series (eq. (3.15)).

gure 3.16). These boxplots are the results of 50 random walks for each length. The inter-quartile range of the left boxplot in Figure 3.16 is determined by 23% to 36% of the random walk length. The median nearly divides the box with a value of 27.5. Compared to the number of non-dominated solutions within the random walks over the 3D-MOML (Figure 3.8), the 4D-MOML achieves about approximately 53% more non-dominated solutions. An hypothesis about the reason for this observation is that the front diversity is significantly lower than in the case of the 3D-MOML. An increase of the random walk length from 100 to 500 (right boxplot of Figure 3.16) results in an increase of the non-dominated solutions about approximately 84.2%. This increase is comparable to the observed increase in the case of the 3D-MOML with 83.9%. The hypothesis mentioned above is verified by the investigation of the front diversity (Figure 3.17). The front numbers 5 and 7 determine the inter-quartile range of the left boxplot in Figure 3.17. The median divides the box exactly with the values 6. Compared to the front diversity of the random walks over the 3D-MOML (Figure 3.9), a decrease of the front diversity about approximately 30% is observable. The increase of the random walk length from 100 to 500 (right boxplot in Figure 3.17) results in a front diversity increase of approximately 52.3%. This percentage increase is only a half of the average increase observed

in the 3D-MOML. This is a logical consequence of the fact that the average number of non-dominated solutions in the random walks of a length of 100 is significantly higher than in the case of the 3D-MOML, but the increase of the non-dominated solution number by an increase of the random walk length is comparable. Therefore, the increase of the front diversity by an increase of the random walk length is considerably lower.
The spread of non-dominated solution diversity in 50 random walks of the length 100 and 500 are depicted in Figure 3.18. In the case of the random walks of the length 100 (left boxplot of Figure 3.18), the spread of the inter-quartile range is comparable to the spread in the random walks of the length 500 (right boxplot of Figure 3.18). In the latter case, the average increase is only slight with a value of 14.4%. This indicates that the level of spread provided by the non-dominated solution is comparable for both time series of different length.
Figure 3.19 depicts the spread of the average rational beeline between each consecutive non-dominated solution over 30 random walks of the length 100 and 500 respectively. In the case of the left boxplot, the size of the inter-quartile range reveals that some of the non-dominated solutions are clustered and others are positioned in a wide range of distances: 50% of the average relational beeline values are between 0.58 and 1.7, which indicates that the distance between the corresponding consecutive non-dominated solutions is more than a half and up to more than 1.5 of the average distance between all consecutive solutions of the time series. The median nearly divides the box with a value of 1, which corresponds to a distance that is equal to the average distance between all consecutive solutions of the random walk. However, the left boxplot in Figure 3.19 reveals some outliers up to a value of 10. This indicates that the distances between the non-dominated solutions are partly considerably higher than in the case of the 3D-MOML. The comparison of the left and the right boxplot reveals that an increase of the random walk length results in a significant decrease of the relational beeline between the non-dominated solutions. The number of outliers is even higher in the case of the random walks of the length 500 compared to those of the length 100. These results are comparable to those of the 3D-MOP. Furthermore, the comparison of the left boxplot to the corresponding results of the 3D-MOML analysis (Figure 3.11) reveals that the upper quartile and even the maximum remains under the level of the average relational beeline values in the case of the 3D-MOML. Otherwise, the relational beeline results of the random walks of the length 500 are nearly comparable in the case of the 3D- and 4D-MOML, indicating that the non-dominated solutions in both MOMLs are mainly clustered.

The ruggedness of the 4D-MOML is further quantified by the investigation

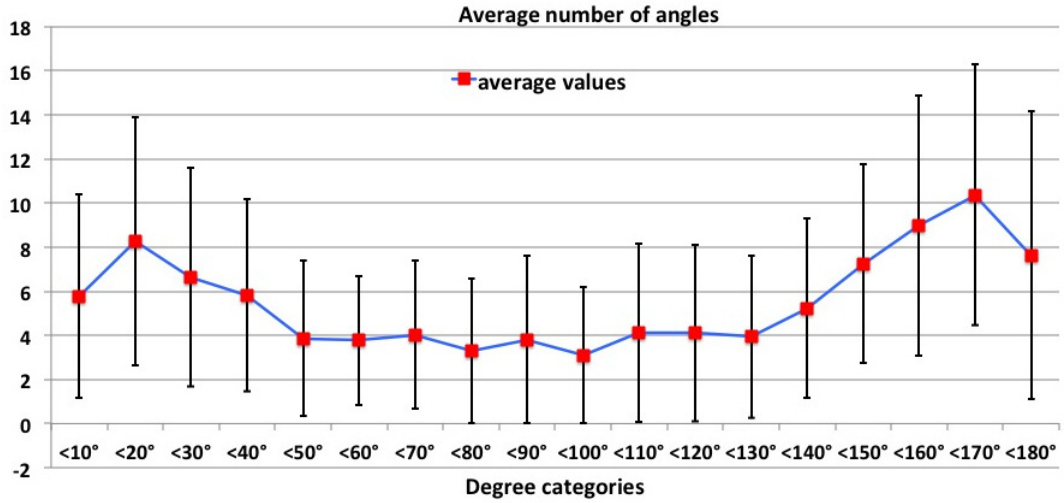

Fig. 3.20: Average number of angles between two consecutive difference vectors categorized in degree intervals of the length 10.

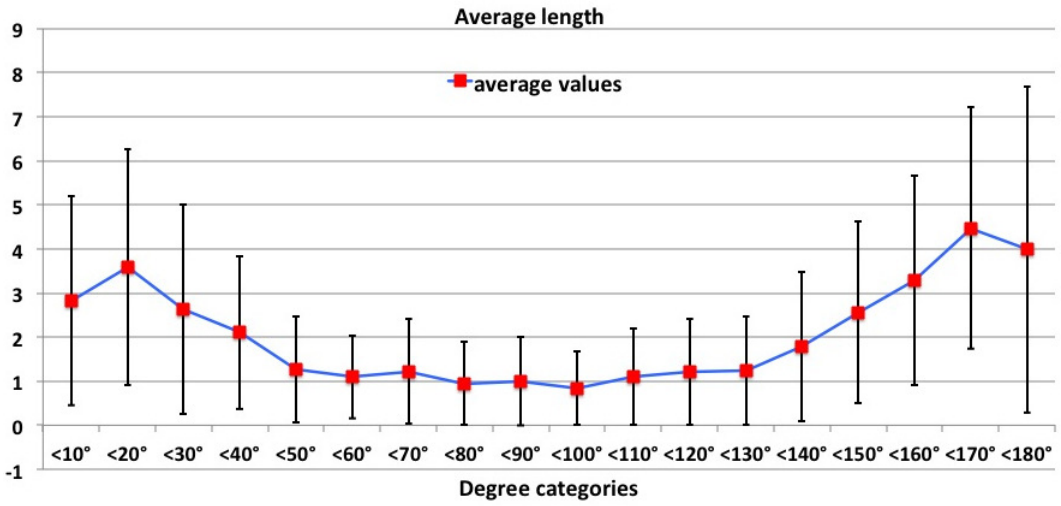

Fig. 3.21: Average length of each two consecutive difference vectors enclosing a particular angle (eq. (3.15)). The angles are categorized in degree intervals of the length 10.

of the difference vectors determined of the consecutive solutions of 30 times series with a length of 100. Firstly, the average number of angles between these difference vectors are determined and presented in Figure 3.20, where the angles are categorized in intervals of the length 10 and depicted on the x-axis. The upper and lower boundaries highlight once again the 95%-significance intervals. The highest angles number is achieved in the interval of $[160°, 170°)$. In this case, the difference vectors are nearly oppositely directed and the single molecular functions are moving (increasing, decreasing and stagnating) very

differently over three time series steps. By way of example: The value of two single molecular functions are oscillating, meaning the function values increase or decrease from a time series step to the succeeding one and are moving the other way round afterwards. The values of the other two single molecular functions increase and decrease or stagnate respectively over the three steps.
In general, higher angle values are more probable than lower ones in the time series, which signals a very rugged landscape. Nevertheless, a local maximum is achieved in the interval of $[10°, 20°)$, which reveals that a considerable number of difference vectors are nearly similarly directed and a great amount of the single molecular functions are moving in the same direction. By way of example: The values of three single molecular functions are increasing or decreasing and stagnating simultaneously over three steps of the time series. The values of the other single molecular function are oscillating. The number of the angles in the intervals of $[30°; 150°)$ are slightly oscillating on a low level. As in the case of the 3D-MOML, the larger the angles, the larger the number of objective functions revealing oscillating behavior over three time series steps.
The comparison of the average angle number distribution over the angles categories of the 3D-MOML (Figure 3.12) and the 4D-MOML (Figure 3.20) reveals that in both cases the probability of high angle values ($> 160°$) and lower angle values ($< 20°$) are significantly higher than the angles values of the other categories. The increase of the MOML dimension results in a shift of the maxima in the direction of the central angles category. This is the consequence of the fact that the probability of these four objective functions moving similarly or oscillating simultaneously is lower than for three objective functions. Furthermore, the lower probability for a similar or oscillating moving behavior of the four molecular functions results in a more even distribution of the angle number over the angle categories.
The investigation of the average path length with particular bending provided by the difference vectors which enclose particular angles categorized once more in intervals of 10 reveals a similar pattern (Figure 3.21): The highest length is achieved for high angle values in the interval $[160°; 170°)$. In general, higher lengths are more probable in the case of high angles values. This indicates large differences of the single molecular function values moving differently. The second highest length is achieved for angle values in the interval $[20°; 30°)$ indicating large differences between the movements of the single molecular function values, which are mainly positioned in the same direction. Slight changes of the single molecular function values are revealed in the interval $[40°; 150°)$ with the lowest length.
The comparison between the results of the average path length to the cor-

responding average number of angles in the case of the 3D- and 4D-MOML reveals similar shapes. This indicates that usually large differences between the single molecular function values of three time series steps are either positioned in different directions or nearly in the same directions.

The 4D-MOML is further investigated according to plateaus, which provides

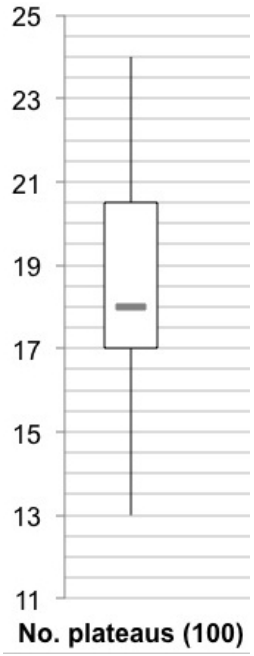

Fig. 3.22: Average number of plateaus characterized by consecutive time series steps of the length 100 in the same Pareto front.

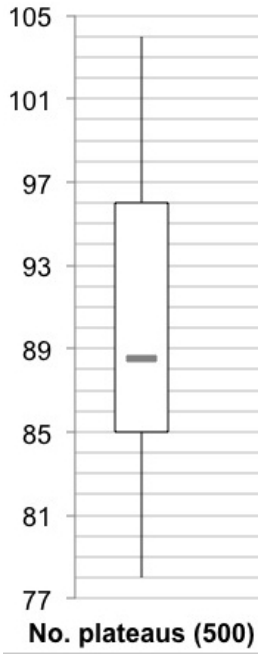

Fig. 3.23: Average number of plateaus characterized by consecutive time series steps of the length 500 in the same Pareto front.

some information about clustered similar qualified solutions. In the first step, the number of plateaus identified by consecutive equal of nearly equal fitness values for each of the four molecular functions are determined (see eq. 3.22)). In 30 random walks of the length 100, eight plateaus have been detected. These plateaus are of a size of only two consecutive (nearly) equal solutions and only on times series comprises two of these plateaus.

In the second step, the number and size of plateaus in the more globally sense of multiple consecutive time series steps assigned to the same Pareto front are presented: Figures 3.22 and 3.23 depict the spread of the average number of plateaus in the time series of the length 100 (Figure 3.22) and 500 (Figure 3.23). 50% of the plateaus numbers in the time series of the length 100 are in a range of 17 to 20.5 with a median of 18. This result reveals a moderate increase of approximately 19% compared to the corresponding results of the 3D-MOML. An increase of the time series length results in an increase of the plateau number of approximately 396%. This percentage increase is significant-

ly higher than the corresponding value of 264% in the case of the 3D-MOML and is a consequence of the considerably lower front diversity of the 4D-MOML compared to the results of the 3D-MOML, as described above (Figure 3.17). The percentage of the detected plateaus assigned to the first front is 36.5%, which is also significantly higher compared the corresponding value of 14.5% in the case of the 3D-MOML. The percentage of the plateaus assigned to the first front in the time series of the length 500 is 12.9%. This value is marginally higher than the corresponding value in the case of the 3D-MOML with 7%. These percentage increases are also a consequence of the lower front diversity.
The average size of these plateaus is determined by the average number of consecutive time series steps belonging to the same Pareto front: The average plateau size is decreased of 3.04 to 2.42 by the increase of the time series length with a standard deviation of 1.63 and 0.83 respectively. The average plateaus size of the first front plateaus is on average larger with 3.08 and 2.75 (standard deviation of 1.42 and 1.29 respectively) determined of the time series of the length 100 and 500 respectively. As the number of non-dominated solutions increases and the front diversity is reduced at the same time compared to the results of the 3D-MOML, the average plateau size is accordingly higher. As in the case of 3D-MOML, the plateaus are unevenly distributed along the random walks.
The final step of the MOML analysis is the direct comparison and visualization of the non-dominated solutions achieved by the 3D- and 4D-MOP in a random walk. Figure 3.24 depicts the objective values of the non-dominated solutions identified in a random walk of the length 100 over the 3D-MOML. Figure 3.25 presents the objective values of the non-dominated solutions identified in the same random walk evaluated by means of the 4D-MOP. The blue path in Figure 3.25 highlights the non-dominated solutions that have also been identified as non-dominated solutions in the case of the 3D-MOP, the red path are further non-dominated solutions, which are not non-dominated in the case of the 3D-MOP. All 10 solutions of the 3D-MOP have also been detected in the case of the 4D-MOP. These two figures exemplarily presents the coherence between the achieved non-dominated solutions of the 3D- and 4D-MOP from the same random walk: usually all 3D-non-dominated solutions are also non-dominated in the case of the 4D-MOP, but the latter one detects some more non-dominated solutions, which are not non-dominated in the case of the 3D-MOP.

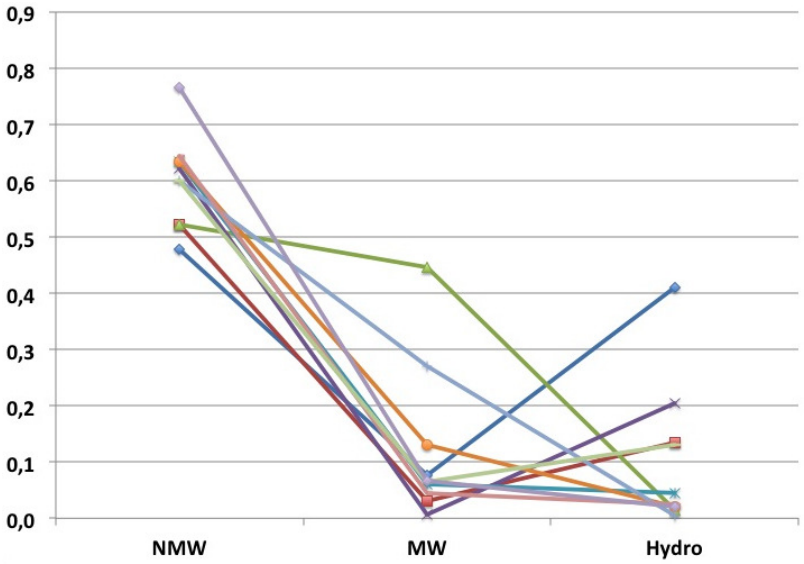

Fig. 3.24: Objective value paths of 10 non-dominated solutions identified in a random walk of the length 100 over the 3D-MOML.

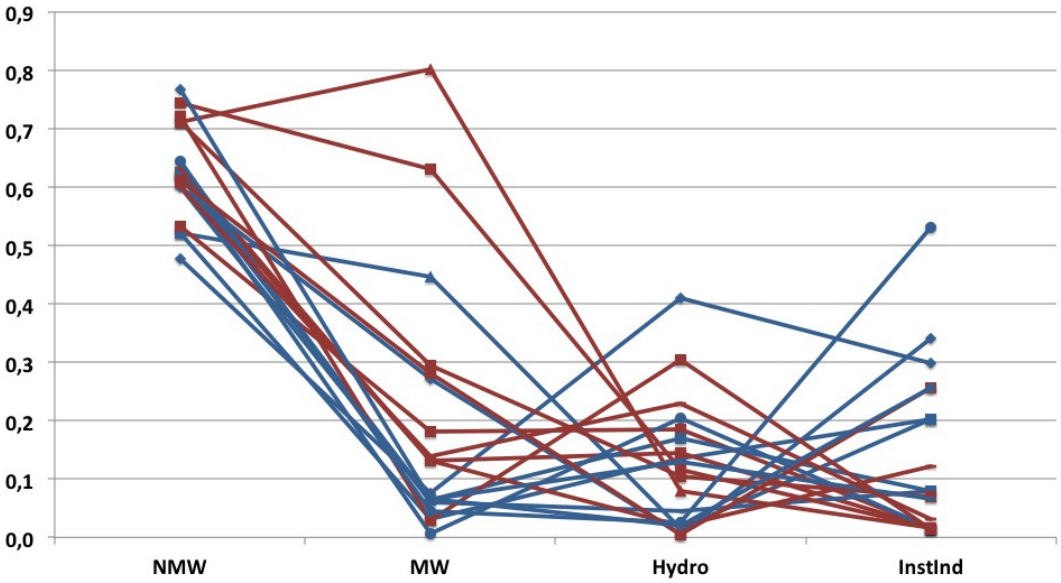

Fig. 3.25: Objective value paths of 19 non-dominated solutions identified in a random walk of the length 100 over the 4D-MOML.

3.8 Consequences of MOML for MOEA Design

The results of the 3D- and 4D-MOML analysis provide some important hints regarding the design of a MOEA. Both, the 3D- and 4D-MOML are very rugged and no specific structure is discernible according to the distribution of non-dominated solutions over the investigated parts of these landscapes of different sizes. The 3D-MOML reveals a higher front diversity and therefore fewer solutions are in the optimal front compared to the 4D-MOML. These results

point out the known fact and challenge for domination-based MOEA that the number of non-dominated solutions increases exponentially with the problem dimension [144]. This observation is thus valid in the case of the proposed 3D- and 4D-MOP and the design of a MOEA has to take account of this fact.
Due to the higher number of non-dominated solutions and the lower front diversity in the case of the 4D-MOP, a far-reaching differentiation of the non-dominated solutions is required. The most intuitive way to perform this differentiation is by assistance of the selection procedure. A selection procedure only based on the non-dominated sorting of the solutions does not provide enough differentiation and as a consequence, a further selection criterion is challenging for this purpose.
Moreover, the 4D-MOP has a higher number of front-based plateaus compared to the 3D-MOP, but this average front-based plateau size of the first front is accordingly smaller. The number of plateaus identified by consecutive equal or nearly equal fitness values for each molecular function is lower in the case of the 4D-MOP compared to the 3D-MOP. The existence of this considerable number of front-based plateaus with approximately 10% first front plateaus in both MOMLs suggests the common approach to balance the search behavior of a MOEA towards exploration in early generations and exploitation in later generations. Thus, variation operators of the MOEA have to support a global search in the first generations of the MOEA to tap potentially high quality solutions, spread over the landscape. In the later generations, a more local search behavior of the MOML supports the search process in the neighborhood of the previously detected high quality solutions.
The increase of the time series length and therefore of the investigated MOML does not result in a proportional increase of non-dominated solutions neither in the case of the 3D-, nor in the case of the 4D-MOML. Moreover, the non-dominated solutions are unevenly distributed over the search space. These facts allow some considerations with regard to the population size: A large population size increases the probability to detect high quality solutions, especially in a very rugged landscape. Therefore, the search performance benefits from a high population size in early generations, but a high population size is counterproductive in later generations, since the probability for the selection of already detected high quality solutions into the succeeding generation decreases with the population size.

4 Design of a MOEA for Peptide Optimization

4.1 Exact Methods versus Metaheuristics

The justification of the design and the use of a metaheuristic to solve the proposed 3D- and 4D-MOP requires a calculation of the runtime complexity to solve these problems exactly. The general advantage of an exact solution method is the calculation of the optimal solutions as opposed to approximative compromise solutions in the case of metaheuristics. The runtime complexity of the exact methods is reducible by the exclusion in advance of some of the feasible solutions based on theoretical considerations. The search space complexity in the present 3D- and 4D-MOP is 20^{20} due to the short peptide sequences of the length 20, composed of the 20 canonical amino acids.
Theoretical considerations of the feasible solutions usually allow the exclusion of different solution categories. The *a priori* exclusion of peptides depends on the application field of peptide optimization. The proposed 3D- and 4D-MOP are selected with the aim of being as generic as possible regarding the determination of their physiochemical properties. Therefore, any *a priori* exclusion is difficult without a concrete application area. Generally, in the field of drug design, peptides have to fulfill the essential properties of being synthesizable and soluble in aqueous solutions. A general guideline is given for the solubility of a peptide in aqueous solutions regarding its primary structure: hydrophobic peptides containing at least 50% hydrophobic residue (A, F, I, L, M, P, V, W, Y)[1] are potentially insoluble or only partly soluble. The number of peptides comprising 20 canonical amino acids of which at least 50% are hydrophobic is

$$\sum_{i=10}^{20} \binom{20}{i} \cdot 9^i \cdot 11^{(20-i)} \approx 4.285x10^{25}, \tag{4.1}$$

[1] `http://www.anaspec.com/content/pdfs/PeptidesolubilityguidelinesFinal.pdf`

where $\binom{20}{i}$ is the number of possible orderings of the hydrophobic amino acids on i of 20 positions of the peptide, 9^i is the number of possible orderings of the 9 hydrophobic amino acids on i positions and $11^{(20-i)}$ are the number of possible orderings of the remaining amino acids on the complementary positions of the peptide. This reduces the search space only slightly to $20^{20} - 4.285x10^{25} \approx 6.2x10^{25}$. Such guidelines as those, for solubility in aqueous solutions, do not exist for the synthesizability by today. Therefore, an exclusion of potentially not synthesizable peptides is not possible. Instead of the exclusion based on theoretical considerations without empirical verification, it is more advisable to take the preferred properties as objective functions and therefore as a part of the molecular optimization problem.
An exact solution of the 3D- or 4D-MOP requires the evaluation of the objective functions MW, NMW, hydro and InstInd for each peptide followed by fast non-dominated sorting. As the computational complexity of NMW is the highest of the objective functions (section 3.4), the complexity of the objective function evaluation is approximately $O(N \cdot l^2)$, where l is the peptide length and N the number of feasible peptides. The following fast non-dominated sorting has a computational complexity of $O(k \cdot N^2)$, where k is the number of objective functions. Even for the 3D-MOP, the computational complexity is $O(N \cdot l^2) + O(k \cdot N^2) \geq 6.2x10^{25} \cdot 20^2 + (6.2x10^{25})^2 \cdot 3 = 1.15x10^{52}$. Assuming the use of world's top soft computer Tianhe-2 developed by China's National University of Defense Technology[2], which performs $30,86x10^{15}$ floating operations per second, this leads to a runtime of $3.74x10^{35} sec. \hat{=} 1.19x10^{28}$ years.

4.2 Introduction to VONSEA

The selection of a MOEA with the focus on customization for molecular optimization is guided by the following expectations: A large number of high quality peptides have to be found within a very low number of generations (less than 20 generations) and within a limited range of population size. The idea of early convergence is motivated by the statement of Singh et al. [145] that each successive generation of an EA provides a progress by discovering high qualified leads in each generation. The specification of a MOEA with the described property by Singh is in the focus of this thesis.
The most established MOEA - NSGA-II - is selected for customization based

[2] http://www.techtimes.com/articles/20429/20141118/worlds-fastest-supercomputer-tianhe-2-is-still-no-1-a-year-later.htm

firstly on the related work of evolutionary peptide optimization as presented in section 2.2. Secondly, the attractiveness of NSGA-II to the present is caused by its simple and intuitive optimization cycle according to an evolutionary process. These advantages are the elaborated design of the NSGA-II operations [4], the usability in many real-world applications ([48], [169], [24]) and its excellent performance in most test problems. Moreover, NSGA-II is very popular as it provides good convergence rates to the Pareto optimal front as well as a good spread of solutions [44]. In the following, the procedure and specificity of NSGA-II is presented. These results justify the design of a metaheuristic to optimize the 3D- and 4D-MOP.

4.2.1 NSGA-II

NSGA-II [44] is an improved MOGA variant of NSGA [143]. The motivation for the evolution of NSGA-II lies in the three drawbacks of NSGA [44]:

1. The non-dominating sorting of NSGA-II as fitness assignment has a high computational complexity of $O(MN^3)$, where M is the number of objectives and N the population size.

2. NSGA provides no elitism. It has been shown that elitism can speed up the performance of a MOEA significantly [181]. Moreover, it prevents the loss of previously found good solutions.

3. NSGA makes use of a sharing parameter. This is part of the diversity preserving mechanism. The sharing parameter is user-defined and influences the performance. Fonseca proposed a dynamic sizing of this parameter [66].

In general, NSGA-II inherits two special features of the MOEA variants MOGA, NSGA and NPGA (Table 1.1): fitness assignment on the basis of non-dominating sorting and the diversity preservation among solutions of the same front. In the following, the advanced components within NSGA-II compared to NSGA are presented [44]:
Firstly, the non-dominated sorting of NSGA-II is improved with the focus on lower computational complexity. This customized sorting starts with the determination of n_p, which denotes the number of solutions dominated by a solution p. Then, the solution set S_p is constructed, which contains the solutions dominated by p. These two steps require a computational complexity of $O(MN^2)$ for comparison. All solutions with $n_p = 0$ are stored in a list F_1 defined as

the current front. The number n_q is reduced by one for each solution q in the set S_p. After that, the solutions with $n_q = 0$ are stored in a separate list Q. After all solutions of the current front have been examined, F_1 is defined as the first front and the process continues with Q as the new current front. The overall computational complexity of this non-dominated sorting is reduced to $O(MN^2)$, but the storage requirement is $O(N^2)$. The procedure of the fast non-dominated sorting is described in the pseudo-code of Algorithm 1 [44].
Secondly, the diversity preserving sharing parameter is replaced by a crowding comparison operator. The definition of this operator makes use of a crowding distance value that is assigned to each solution. For this purpose, a sorting of the population according to each objective function value in ascending order is required. The crowding distance value $i_{distance}$ of boundary solutions for each objective function is assigned to infinity. The crowding distance operator is part of the solution process and is responsible for the uniform spread of the solutions on the Pareto front. The infinity assignment to boundary solutions ensures that these points are always selected. For the other intermediate solutions, $i_{distance}$ is determined by the average distance of the two solutions on either side along each of the objective. The overall crowding distance of the intermediate solution i is calculated by:

$$I[i]_{distance} = I[i]_{distance} + \frac{I[i+1]_m - I[i-1]_m}{f_m^{max} - f_m^{min}} \tag{4.2}$$

where $I[i]_m$ refers to the m-th objective function value of the i-th solution and $I[i]_{distance}$ is initialized with 0. f_m^{max} and f_m^{min} are the maximal and the minimal value of the m-th objective function. The crowding distance assignment has a computational complexity of $O(MN\frac{1}{2}logN)$, which is governed by the sorting of the objective function values in an ascending order to assign the distance values. The computational complexity is calculated on the basis of M independent sorts of at most N solutions in the case that all solutions are in the same front. The crowding comparison operator $\prec_n$ is defined on the basis of the two attributes i_{rank} and $i_{distance}$ of a solution i, where $i_{distance}$ is a measure for the perimeter of the cuboid that is spanned by the two neighboring solutions on either side of solution i, and i_{rank} is the rank of solution i. The crowding comparison operator is defined as a partial order that prefers the solution i with a lower (and better) rank to a solution j. In the case that the two solutions i and j are in the same front, the solution in the less crowded area is preferred: $i \prec_n j$ ***if*** $(i_{rank} < j_{rank})$ ***or*** $((i_{rank} = j_{rank})$ *and* $(i_{distance} > j_{distance}))$. [44]
The sorting on $\prec_n$ has a computational complexity of $O(2N \cdot log(2N))$. [44]
Finally, binary tournament selection (introduced in section 2.3.4.1) is used as

Algorithm 1: Pseudocode: Non-dominated sorting of population P

```
1  foreach p ∈ P: do
2  |  initialize S_p = ∅, n_p = 0
3  |  foreach q ∈ P: do
4  |  |  if (p ≺ q) then S_p = S_p ∪ {q} ;        // If p dominates q,
   |  |                  // add q to the solution set dominated by p
5  |  |  ;
6  |  |  else if (q ≺ p) then
7  |  |  |  n_p = n_p + 1;
8  |  |  end
9  |  end
10 |  if n_p = 0 then
11 |  |  p_rank = 1;
12 |  |  F_1 = F_1 ∪ {p};              // p belongs to the first front
13 |  end
14 end
15 initialize the front counter: i = 1:
16 while F_i ≠ ∅ do
17 |  Q = ∅;              // Q stores the members of the next front
18 |  foreach p ∈ F_i do
19 |  |  foreach q ∈ S_p do
20 |  |  |  n_q = n_q − 1;
21 |  |  |  if n_q = 0 then
22 |  |  |  |  q_rank = i + 1;
23 |  |  |  |  Q = Q ∪ {q};          // q belongs to the next front
24 |  |  |  end
25 |  |  end
26 |  end
27 |  i = i + 1;
28 |  F_i = Q;
29 end
```

a selection strategy within NSGA-II. The selection criterion is based on the crowding comparison operator $\prec_n$.

Procedure of NSGA-II

1. **Initialization of the start population**

   a) Initialization: Start population P_0 is randomly initialized with size N, the generation counter is set to $t = 0$.

   b) Ranking: Non-dominated sorting of the individuals into fronts. A fitness value is assigned to each individual equal to its front[3].

   c) Creation of the offspring population Q_0 of size N.
   while $|Q_0| < N$:

      i. Binary tournament selection: Selection according to the rank and the crowding distance.

      ii. Recombination and mutation

2. **Main loop**

   a) Combination of parent and offspring set: $R_t = P_t \cup Q_t$

   b) Ranking: Non-dominated sorting is applied on R_t into the fronts F_i. A fitness value is assigned to each individual equal to its front.

   c) Set $P_{t+1} = \{\ \}$, i=1.
   while ($|P_{t+1}| + |F_i| < N$) {

      crowding distance assignment in F_i,

      $P_{t+1} = P_{t+1} \cup F_i$,

      $i = i + 1$, }

      Sorting of $(F_i, \prec_n)$ in ascending order,
      $P_{t+1} = P_{t+1} \cup F_i'$ with $F_i' \subset F_i$ such that $|P_{t+1}| + |F_i'| = N$.

   d) Creation of the succeeding population Q_{t+1} of size N.
   until $|Q_{t+1}| < N$:

[3]The front is the non-domination level: front 1 is the front with the best or non-dominated solutions, front 2 is the next best level, and so on.

i. Binary tournament selection: Selection according to the rank and the crowding distance.

ii. Recombination and mutation

e) $t = t + 1$

f) **if** $t == T$: *STOP*,
else repeat the main loop

The procedure of NSGA-II starts in step 1.a) with the random initialization of the start population P_0. In step 1.b), the individuals of the start population are ranked into fronts, where the first front contains all non-dominated solutions. The offspring population Q_0 of size N is determined in step 1.c). Therefore, the individuals for reproduction are selected by binary tournament selection according to the rank and the crowding distance of the individuals. Two selected individuals are recombined and mutated to create two offspring. The main loop starts in step 2.a) with the combination of the parent population P_0 and the offspring population Q_0 to the population R_0 of size $2N$. Step 2.b) is the repetition of step 1.b) applied on R_0. The succeeding population is created in step 2.c), where the population P_1 is filled with the N-best individuals of R_0 according to the rank and the crowding distance. The succeeding offspring population Q_1 is determined in step 2.d), which is a repetition of step 1.c) applied on P_1. If the total number of generations T is achieved, the main loop stops, otherwise the main loop continues with the combination of the current population P_1 and the offspring population Q_1.
A well-known disadvantage of NSGA-II is the fact that the performance of NSGA-II worsens with an increase of the objective number, more precisely for more than three objectives [98], [92]. The main reason for this is the increasing number of solutions with $i_{rank} = 1$ or the number of non-dominated solutions in the population at an early search stage compromising the convergence properties. Instead of the standard Pareto dominance, Sato [135] proposed a modified dominance definition that clearly improves the performance of NSGA-II for many-objectives problems (more than three objectives). Another alternative is the replacement of the crowding distance operator.
A lot of research has been done to adapt NSGA-II for different improvements: D'Souza et al. improved the NSGA-II by the principle of space-time trade-off in the non-dominated sorting stage to reduce run-time complexity and to improve convergence [48]. The performance of the modified NSGA-II is tested on the classification problem of leukemia based on microarray data. Fang

et al. adapted the NSGA-II to improve run-time complexity [62]. The computational complexity of the non-dominated sorting algorithm - as used in the NSGA-II - is reduced by reducing the number of redundant comparisons, which arises by the recording of the dominance information among solutions from their first comparison. A new data structure termed dominance tree and the divide-and-conquer mechanism are introduced. Tran introduced a MOEA termed Adaptable NSGA-II (ANSGA-II) which overcomes the problem of required parameter tuning to achieve a good performance for an arbitrarily complex problem [155]. ANSGA-II includes an adaptive population size as well as a self-adaptive crossover and mutation operator. Li et al. adapted the NSGA-II in the crowding distance method, which is designed by the minimum spanning tree to improve the diversity of the solutions [106]. The performance is tested on two and three objective test problems and reveals a comparative convergence and a good diversity performance compared to the NSGA-II. A hybridization of NSGA-II is presented by Bechikh et al. [15]. Pareto Hill Climbing NSGA-II (PHC-NSGA-II) provides the specific local search procedure Pareto Hill Climbing. The aim of PHC-NSGA-II is to enhance the convergence rates and the spread of the solutions by improving the search pattern. An adaptive mutation variant for NSGA-II is introduced by Carvalho and Araujo [26] with the same goal of improving diversity and convergence. The mutation rates are controlled by using information about the diversity of the candidate solutions. Fortin and Parizeau [68] address the instability of the crowding distance that appears in the case of two or more solutions sharing the same fitness. The bias induced by the individuals sharing the same fitness during the selection process is fixed by crowding distance computation with unique fitnesses. Furthermore, unique fitness based tournament selection is used to select the parent solutions. Studies performed with two-objective optimization problems are evaluated with regard to the influence of the adapted binary tournament selection on the performance. The convergence is improved by the additional selection pressure, but the diversity is unaffected by this adaption. Deb and Jain presented the reference-point based Many-Objective NSGA-II (MO-NSGA-II) [40]. MO-NSGA-II emphasizes non-dominated solutions close to a set of well-distributed reference points. Experiments have been performed on three to 10 objectives in comparison to the performance of MOEA/D, where MO-NSGA-II revealed superior performance. Seada and Deb recently proposed the unified EA termed U-NSGA-II, especially developed for many-objective problems [144]. U-NSGA-II performs well on mono- as well as multi- and many-objective problems. It uses a set of reference points as well as a niching-based selection operator.

4.2.2 The procedure of VONSEA

NSGA-II as described in the last section is customized in its components with the main goal of molecular optimization under the special expectation of early convergence. This customized NSGA-II is termed Variation Operator specific Non-dominated Sorting EA (VONSEA). The more general denotation 'EA' refers to the fact that the proposed algorithm is in general a customization of NSGA-II, but comprises ES components and strategies. More precisely, the role of the component selection strategy as well as the types of the variation operators within VONSEA are characteristic for ES, but unfamiliar in the area of GA: Firstly, the selection process within VONSEA is not used to determine the individuals for reproduction. Instead of using the binary tournament selection according to the rank and the crowding distance, a sophisticated selection strategy is applied to determine the succeeding population. The use of a selection process as a method for the determination of the succeeding population is the characteristic role of the selection within the area of ES. Secondly, the denotation 'Variation Operator Specific' reflects the opportunity to choose between several recombination and mutation variants. Some of these variation operators have been either evolved in the area of ES or are designed according to these models.

The procedure of VONSEA is depicted in the flow chart (Fig. 4.1) and only the parts differing from the NSGA-II procedure - as presented in the last section - are mentioned. A detailed description of these components and the motivation for their evolution or the parameter settings are given in the following chapter. The individuals for reproduction are selected by Stochastic Universal Sampling (SUS) instead of binary tournament selection according to the rank and the crowding distance as used within NSGA-II. An individual is allowed to be selected multiple times. The individuals of the population are assigned to different segments representing the Pareto fronts. The front-based SUS is implemented in VONSEA as illustrated in the pseudocode of Algorithm 2.

Three individuals are selected as parents for recombination and mutation. Therefore, the number of pointers is set to three (line 1). The individuals of the population are ranked into front sets by fast non-dominated sorting (line 2). Figuratively speaking, the front sets are arranged on the roulette wheel according to their front set size (line 3). The first pointer is assigned to the wheel by a random number in the interval $[0, 1/numberOfPointers)$ (line 5) from the segment forefront of the front set with the largest size. The pointers are positioned by the distance from the first pointer (line 7). The indexes of the individuals associated to the pointers are determined in line 8. The individuals

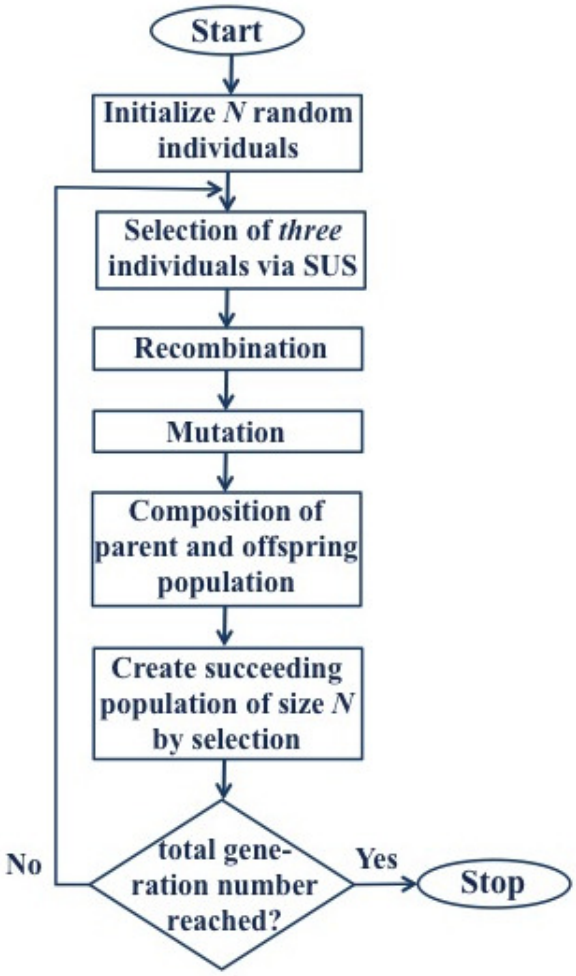

Fig. 4.1: Procedure of the Variation Specific Non-dominated Sorting Evolutionary Algorithm (VONSEA)

selected by these pointers are added to the parent population (line 9).
Additionally, VONSEA differs from NSGA-II in the components recombination and mutation. Optionally, different recombination and mutation operators are implemented to compose a VONSEA configuration instead of the default variation operators of NSGA-II. Furthermore, VONSEA differs in the procedure to determine the succeeding population Q_t. Instead of ranking the composed population of parents and children and selecting the N best individuals, three kinds of selection strategy are proposed that use a combination of fitness-proportionate selection and a discerning selection criterion, which is front-based in one case and indicator-based in the other. These components are motivated and described in the following sections in more detail.

Algorithm 2: Pseudocode: front-based SUS in VONSEA

```
1  numberOfPointers = 3;
2 FastNondominatedSorting(population, fronts);
3 FrontSortingMagnitude(population);
4 distance=(numberOfPointers)^-1;
5 ptr(0)= randomNumber · distance;
6 for i=0, ..., numberOfPointers-1 do
7 |  ptr(i+1) = (int) (ptr(0)+ i · distance);
8 |  index = ptr(i+1) · population_size;
9 |  parent_population.add(population.getIndividuals(ptr(index)));
  |
10 end
```

4.3 Encoding Scheme of VONSEA

4.3.1 Encoding Principles

A GA works on two types of spaces: the coding space (genotype) and the solution space (phenotype). The encoding of the genotype is important from two points of view: Firstly, a genotype-phenotype encoding for the individual representation necessitates the use of a transformation or decoding to map a variable from the coding space into the solution space for fitness evaluation and selection. Secondly, the GA components recombination and mutation have to be developed according to the encoding as these operators directly work on the variables of the coding space: It has been shown that a special selection of recombination and mutation operators work well on different types of encoding [82]. Furthermore, the encoding has a strong impact on the performance of a GA and has to be chosen carefully. (see e.g. [3], [1]) Goldberg was one of the first who presented recommendations for the construction of binary representations. He formulates two abstract and general principles for the design of representations on the basis of theoretical schemata considerations [75]:

- **The principle of building blocks:** The encoding scheme has to be short, of low order and is relatively unrelated to schemata over other fixed positions. This prevents a long and high ordered scheme being disrupted by recombination and mutation.
- **The principle of minimal alphabet:** The encoding scheme has to be

as small as possible while still allowing a natural representation of the variables.

The principle of meaningful building blocks is motivated by the scheme theorem [21]. The principle of minimal alphabet advises the increase of the potential number of schemata by reducing the cardinality of the alphabet. The principles are provided for binary strings and are not advisable as design criteria for non-binary strings. Kershenbaum formulates more precise and applicable guidelines. The recommendations are originally tailored for tree representations, but they are also applicable to other representation types. Five possibly conflicting properties are advised for an ideal encoding [91]:

1. The encoding scheme has to represent all feasible solutions.
2. The encoding scheme has to represent only feasible solutions.
3. All feasible solutions have an equal probability of being represented.
4. The encoding scheme has to represent a useful scheme in a small number of genes that are close to one another in the chromosome.
5. The encoding scheme has to possess locality in the way that small changes to the chromosome result in small changes in the solution.

The first recommendation is a property that is usually easily satisfied. The second recommendation sometimes requires a compromise: A small number of infeasible solutions is better than a high number as this increases the probability of creating infeasible solutions by the variation operators and makes a GA ineffective. The third recommendation ensures the creation of diverse random starting solutions. Furthermore, the GA is more effective in exploring the entire solution space. The fourth recommendation is the most difficult property of an encoding and it is generally challenging to develop a suitable encoding according to this property. The property locality in the fifth recommendation refers to a genotype-phenotype mapping. It describes how well neighbored genotypes correspond to neighbors in the phenotype space. This fifth recommendation ensures that the GA is able to perform a guided local search. Otherwise, a low locality results in a more random search instead of a guided search of the GA. Goldberg further classified the encoding schemata referring to the fact that the fitness functions for each encoding scheme depend either on the factor 'value', 'order' or both:

(i) a scheme where fitness depends on order only.

(ii) a scheme where fitness depends on order and value.
(iii) a scheme where fitness depends on value only.

In the following, the most common encoding schemes in GAs for each category are presented:
The best-known encoding scheme of category (*i*) is the **permutation** and it is used in combinatorial optimization problems like the Traveling Salesman Problem [139]. This problem is specified by a list of cities. The goal of the search process is to find a route that visits each city only once and has minimal length. A natural representation is an ordered list of city numbers:

Example 1 ***Individual:*** 1 2 4 3 5 6 9 8 7.

The permutation is used in another application example from bioinformatic [173]. It is used as encoding scheme in a GA to predict the secondary structure of **RNA!** molecules. The secondary structure is encoded as permutation and the GA predicts the specific canonical base pairs that perform hydrogen bonds and build helixes. Specific variation operators are reasonable for permutation encoding: recombination and mutation corrections have to be performed to leave chromosome consistent [100]. The recombination operators associated with permutation are the Partially Mapped Crossover (PMC) [77], the Cycle Crossover (CX) [34] and the Order Crossover (OX) [119]. The mutation operator associated with permutation is the inversion that changes the location of characters. The disadvantage of the recombination operators for permutation encoding is the high implementation complexity of these crossover operators [100].
The most common encoding scheme of category (*ii*) is the **binary encoding**. Each individual is represented as a binary string of the bits 0 and 1. The following example shows a hexadecimal encoding:

Example 2 ***Individual:*** *1101011101101.*

Moreover, each allele represents a value. An advantage of binary encoding is its support of a wide range of recombination operators. Furthermore, it fulfills the design principles of Goldberg [75] best. Binary encoding causes problems in the case of a continuous search space with large dimension [82]: If a variable has a finite number of discrete valid values, some of the binary codes are redundant.
The most common encoding scheme of category (*iii*) is the value encoding. Individuals are represented as strings of some kind of value like integer, real or character:

Example 3 ***Individual (real encoding):*** 1.23 2.54 3.55 6.73 2.12.
Individual (character encoding): *ABDKGWUFEKZWBS*

Real encoding is of increasing interest in the field of real-world optimization problems, like in the field of chemometry [108] or biotechnology [133]. The popularity of real coding is due to the following advantages [82]: Firstly, real coding is very close to the natural representation of the variables for many optimization problems. As there is no difference between genotype and phenotype in these cases, a genotype-phenotype mapping is not necessary. Secondly, real coding is very natural in optimization problems with variables in continuous domains. Thirdly, real coding has the potential to exploit the concept of graduality of the fitness functions with continuous variables. This means that small changes in the variables correspond to small changes in the fitness function values.

4.3.2 Encoding of Individuals in VONSEA

The individuals in VONSEA represent short peptide sequences composed of 20 amino acids. These amino acids are the 20 canonical amino acids as listed in Table 4.1. Three different individual encodings of these peptide sequences within MOEA are conceivable: Firstly, the encoding of a peptide sequence as character string, secondly an encoding of the single amino acids as nucleotide triples and thirdly an encoding of the amino acids by bit strings. These encoding approaches are discussed in the following.
Several tools providing molecular functions to determine physiochemical or structural properties of peptides make use of a single-letter code for the amino acids, depicted in the right column of Table 4.1 (e.g. see [123], [81], amino acids substitution matrices from protein blocks used within the Needleman-Wunsch Algorithm for global sequence alignment). The individuals within VONSEA are encoded as character strings composed of 20 different characters according to the single-letter code to provide the required input structure for the molecular fitness functions and to avoid a transferring into this input structure before the evaluation of every fitness function.

Example 4 ***Individual in VONSEA:*** *ADIHMNLKFPSTVWYRCEQG*

Therefore, this encoding represents a value encoding and is classified in category (*ii*) 'a scheme where fitness depends on order and value' in a broader

Amino Acid	Char code
Alanine	A
Arginine	R
Asparagine	N
Aspartic acid	D
Cystein	C
Glutamic acid	E
Glutamine	Q
Glycine	G
Histidine	H
Isoleucine	I
Leucine	L
Lysine	K
Methionine	M
Phenylalanine	F
Proline	P
Serine	S
Threonine	T
Tryptophan	W
Tyrosine	Y
Valine	V

Table 4.1: List of the 20 canonical amino acids and the established one letter code used for the encoding in VONSEA

sense: In the case of molecular functions predicting peptide properties, each amino acid or character is identified with physiochemical property values and the molecular functions work on these single characters as well as on the ordering of the amino acids (or characters) in a sequence, which is a decisive factor on several peptide properties.
In the following, the properties of the proposed character encoding are summarized and related to the recommendations of Kershenbaum:

- every peptide of the solution space is exactly represented by a character string
- every feasible character string presents exactly one peptide
- all peptides are equally represented

- a genotype-phenotype mapping is not necessary
- small changes performed by a variation operator on the character strings preserve similarity of the created offspring to their parents

The first two properties ensure that all feasible - and only the feasible - solutions are represented by this character encoding. The characters used have an equal probability of being represented in a solution and therefore each feasible solution has the same probability of being represented. The phenotype representation allows a depiction as string with the letter code of the canonical amino acids; therefore a genotype-phenotype mapping is not necessary. The last property allows the assessment whether small changes in the molecule structure is related to similar molecule properties [55]

Another potential encoding scheme of the peptide strings is the presentation

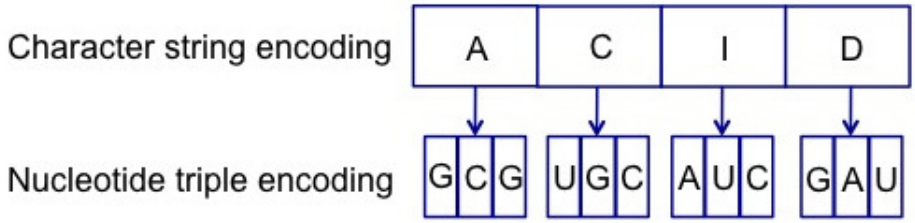

Fig. 4.2: Peptide presentation as character string and nucleotide string encoding.

of the canonical amino acids by code triples. The single amino acids are represented by nucleotide triples consisting of the four nucleotides A, U, G and C. Figure 4.2 depicts the genotype-phenotype representation of a peptide of 5 amino acids and the corresponding code triple encoding. With 4^3, the number of nucleotide combinations is much higher than the number of coded amino acids, thus revealing a high number of unfeasible combinations. Therefore, this encoding scheme has the following undesirable properties:

- the peptides have different representation forms as most of the single amino acids are encodable by differing nucleotide triples (see Table 4.2 in section 4.5)
- a very high number of nucleotide triple encoded peptides are unfeasible
- a genotype-phenotype mapping is necessary
- the peptides have a differing number of representation forms and have a different probability to be presented

Nucleotide triple encoding requires a higher implementation complexity and storage. Therefore, the character string encoding is preferred over nucleotide triple encoding.
A similar approach is the encoding of the single amino acids as bit strings. As 20 amino acids have to be coded, the bit strings require at least a bit string length of at least 5, as the number of bit string combinations provides $2^5 = 32$ possible combinations. According to the advice of Goldberg referring to binary encoding, this encoding scheme is not advisable: The disadvantage of binary encoding for the presented purpose of peptide optimization is the representation of infeasible peptides which is a general disadvantage of binary encoding as mentioned above. An exclusion of these infeasible encodings implies a higher implementation complexity. Compared to nucleotide triple encoding, bit strings encoding also requires a genotype - phenotype mapping. Otherwise, all bit string encoded peptides are equally represented and every feasible bit string peptide represents exactly one peptide.

4.4 Recombination Operators of VONSEA

The use of recombination - also termed crossover - operators in genetic algorithms is motivated by their property of disruption. This encourages the exploration of the search space and makes the search more robust. The disruption is realized by a genetic exchange of information between different individuals chosen as parents. The genetic material of the parents is reassembled in order to produce offspring with a combination of good structures of the parents improving the overall fitness. Recombination is established as a primary search operator (e.g. see [84]).
Recombination theory is mainly developed in the GA area. As an optimization problem specific encoding is established in the GA area, the recombination operator has to be adjusted to the individuals' representation. As a consequence, the conventional recombination operators are classified into binary, floating-point and real-valued types.

4.4.1 Traditional Recombination Operators

Fogel was one of the first to introduce recombination in EAs (e.g. see [64]), otherwise Holland [84] was the first who proposed a theoretical emphasis on recombination. Holland was focused on schemata which provide a basis for pro-

mising attribute combinations with the goal of improving the GA performance. The one-point recombination was the first which was theoretically investigated by Holland [84]. The procedure of this operator is the selection of two random individuals as parents. Then, one random crossover point is selected with uniform probability and the genetic material at this point is exchanged. The key aspect of one-point recombination is the fact that the offspring start with the schematic part of one of the parents and potentially provide a promising building block. The potential of this block in optimizing the performance is tested in new contexts afterwards. Due to this, the disruption of long schemata is much more likely than small ones.
De Jong proposed the analysis of n-point recombination [35]. In contrast to one-point recombination, n random recombination points are selected and the genetic material between these points is exchanged. Compared to the one-point recombination, n-point recombination is less likely to disrupt long schemata.
Syswerda introduced a new form of recombination - the uniform recombination - as a consequence of empirical studies which show an advantage in using more than two crossing points [150]. Uniform crossover creates an offspring by randomly selecting each bit either from the first or the second parent. Syswerda analyzed uniform recombination in comparison to one- and two-point recombinations. In this context, uniform recombination surprisingly has no length bias and its more disruptive nature has been interpreted in the sense that it is more likely to create instances of new high-ordered schemata from lower-ordered ones than the one- or two-point recombinations. Uniform crossover is very effective for some problems in the case of an average of $(L/2)$ crossings (L is the chromosome length) [150]. Another important but more general conclusion of this theoretical analysis is that a broader theory is necessary to indicate a balance between exploration and exploitation by appropriate choices of the population size, genetic operator rates and selection pressure [35]. In the work of Eshelman [58], it is demonstrated that uniform crossover has a more powerful exploration property than n-point crossover. In this work, Eshelman further introduced a characterization of recombinations: recombinations with positional or distributional bias. Recombinations have positional bias if the creation of a new scheme by recombining existing ones depends on the location of the alleles in a chromosome. The positional bias is similar to length bias. Otherwise, a recombination has distributional bias in the case that the amount of material to be exchanged is not uniformly distributed. Furthermore, Eshelman assigns one-point, two-point and uniform recombination to these two characterizations: One-point recombination has high positional bias as all alleles over the chromosome are potentially exchanged with uniform probability of $1/L$. Two-point recombination is of a lower positional but no distributional bias, whereas uniform recombination has no positional but high distributional bias

as the amount of material exchanged is binomially distributed.
Besides one-point, n-point and uniform recombination, shuffle crossover is also classified as a recombination type of binary-encoded GA: Shuffle crossover is related to uniform crossover. One crossing point is randomly chosen, but before the alleles of the two parents are exchanged, both parents are shuffled at random. After the recombination, the alleles of the offspring are shuffled again. This procedure ensures the removal of positional bias [25].
The recombination methods for floating-point encoded GAs are either a straightforward imitation of the recombination types for binary-coded GAs or they are performed by averaging the alleles of each two parents. The disadvantage of the imitation is the fact that only mutation operators are able to insert new values into the population, since the recombination is only able to combine the existing ones in a different manner [122]. Arithmetic recombinations are established in floating-point GAs and are based on the averaging of the parents alleles [113]. There are three versions of arithmetic recombination (an overview is given by Picek et al. [122]): Simple arithmetic recombination chooses k recombination points. Then, the first k float alleles of a randomly selected parent are copied into the offspring. The remaining alleles are the arithmetic averaging of the two parents ${x_i}^p$ and ${x_j}^p$. The two offspring are of the following form:

$${x_i}^{p+1} = \langle {x_{i,1}}^p, {x_{i,2}}^p, ..., a \cdot {x_{j,k+1}}^p + (1-a) \cdot {x_{i,k+1}}^p, ..., a \cdot {x_{j,n}}^p + (1-a) \cdot {x_{i,n}}^p \rangle \tag{4.3}$$

$${x_j}^{p+1} = \langle {x_{j,1}}^p, {x_{j,2}}^p, ..., a \cdot {x_{i,k+1}}^p + (1-a) \cdot {x_{j,k+1}}^p, ..., a \cdot {x_{i,n}}^p + (1-a) \cdot {x_{j,n}}^p \rangle\,, \tag{4.4}$$

where a is the weighting factor with $a \in [0,1]$. There are three variants regarding the choice of a: This factor is constant, it is picked at random in every recombination step or it is a variable that depends on the current generation number.
Single arithmetic recombination differs from the simple one in the averaging of only one randomly selected allele of the two parents. The remaining alleles are copied from the parent:

$${x_i}^{p+1} = \langle {x_{i,1}}^p, {x_{i,2}}^p, ..., a \cdot {x_{j,k}}^p + (1-a) \cdot {x_{i,k}}^p, {x_{i,k+1}}^p, ..., {x_{i,n}}^p \rangle \tag{4.5}$$

$${x_j}^{p+1} = \langle {x_{j,1}}^p, {x_{j,2}}^p, ..., a \cdot {x_{i,k}}^p + (1-a) \cdot {x_{j,k}}^p, {x_{j,k+1}}^p, ..., \cdot {x_{j,n}}^p \rangle\,. \tag{4.6}$$

The most commonly used arithmetic recombination strategy is the 'whole arithmetic crossover'. All alleles of the offspring are calculated by the arithmetic average of all parents alleles.

$$x_i^{p+1} = a \cdot x_i^p + (1+a) \cdot x_j^p \tag{4.7}$$

$$x_j^{p+1} = (1-a) \cdot x_i^p + a \cdot x_j^p, \tag{4.8}$$

In the case of real-coded GAs, two types of categorization are proposed for recombinations. The first category divides the recombination operators in mean-centric and parent-centric operators [42]: In the case of mean-centric operators, the offspring are created around the mean of the participating parents and around one participating parent in the case of parent-centric recombinations. The Unimodal Normal Distribution Crossover (UNDX), the SimPlex Crossover (SPX) and the BLend Crossover (BLX)-α are mean-centric, whereas the Simulated Binary Crossover (SBX) and the Parent-Centric Crossover (PCX) are parent-centric.
Deb et al. present a categorization of recombination operators in variable-wise and vector-wise operators [47]: Variable-wise operators recombine the variables of the participating parents independently from one another. These operators do not take linkages between variables into account. Representatives of this category are BLX-α and SBX. In the case of vector-wise recombination operators, a linear combination of the complete variable vectors of the participating parents is created to produce offspring. These operators take account of the linkages between the variables. Representatives of this category are UNDX, PCX and SPX. The recombination operators UNDX, BLX-α, SPX, SBX, and PCX are described in the following.

4.4.1.1 Mean-centric Recombination Operators

UNDX [120] was proposed by Ono and Kobayashi and creates offspring around the mean center $\vec{g}$ of the participant parents. With a small probability, offspring are created away from this center. UNDX chooses $(\mu-1)$ individuals as parents at random and then the mean vector $\vec{g}$ of these individuals is computed. The offspring is created as follows:

$$\vec{y} = \vec{g} + \sum_{i=1}^{\mu-1} w_i |\vec{d}^{(i)}| \vec{e}^{(i)} + \sum_{i=\mu}^{n} v_i D \vec{e}^{(i)}, \tag{4.9}$$

where w_i and v_i are zero-mean normally distributed random numbers denoted as $N(0, \sigma_1)$ and $N(0, \sigma_2)$. Kita and Yamamura [93] advised values of $\sigma_1 = 1/\sqrt{\mu - 2}$ and $\sigma_2 = 0.35/\sqrt{n - \mu - 2}$, where n is the size of the variable vector. D is the length of $\vec{d}^{\mu+1}$ orthogonal to $\vec{d}^{1}, ..., \vec{d}^{\mu-1}$ with $\vec{d}^{(i)} = \vec{x}^{(i)} - \vec{g}$. $\vec{e}^{(i)} = \vec{d}^{(i)}/|\vec{d}^{(i)}|$ is the orthonormal basis. The goal of this recombination method is that the offspring are created around the mean vector. The probability for producing an offspring far away from its parent is reduced and the maximal probability is assigned at the mean vector. The computational complexity to

produce an offspring is $O(\mu^2)$, mainly caused by Gram-Schmidt orthonormalization process [42].
BLX-α is a variable-wise recombination operator and was introduced by Eshelman and Schaffer [60]. The offspring are uniformly created around the two parent values. Two individuals x_1 and x_2 are selected randomly as parents and two offspring are created. For that purpose, each component x_c^i of an offspring vector x_c is a randomly chosen value from $[x_1^i, x_2^i]$ with

$$x_1^i = min(x_1^i, x_2^i) - \alpha \cdot d_i \tag{4.10}$$

$$x_2^i = max(x_1^i, x_2^i) + \alpha \cdot d_i \tag{4.11}$$

$$d_i = |x_1^i - x_2^i| \tag{4.12}$$

where $x_{1,2}^i$ are the i-th components of $x_{1,2}$ and a positive parameter α. The user-defined parameter α is responsible for the creation of an offspring inside or outside the parent range. Herrera tested different values for α [82]. The value $\alpha = 0.5$ achieved the best performance. The component-wise creation of an offspring causes difficulties in optimizing non-separable fitness functions because of the mutual dependency among the variables that are not well considered [120]. To overcome these difficulties, Eshelman et al. extended the first version to BLX-α-β and BLX-α-β-γ [59]. The parameters are problem-specific and difficult to determine.
SPX is presented by Tsutsui et al. [157]. This operator is an extension of BLX-α. It generates offspring vector values by uniform sampling values from m parent vectors with ($2 \leq m \leq$ number of parameters +1). The goal of this operator is that offspring are created around the mean of the parents, but they are restricted within a predefined region. This region is $\sqrt{\mu + 1}$ times bigger than the parents simplex. Furthermore, the offspring are uniformly distributed over this region. The selection of three parents is advised for low dimensional problems and four parents are advised for higher dimensions [157]. The computational complexity for creating an offspring is about $O(\mu)$ [42].

4.4.1.2 Parent-centric Recombination Operators

The variable-wise operator SBX is presented by Deb and Agarwal [38]. SBX is evolved for real-coded GAs to simulate the effect of one-point recombination in binary-coded GAs. Individuals close to their parents are created with the help of a polynomial probability distribution and two parent values of a particular variable. Two particular parent values $p_{1,2}^i$ of the i-th variable are linearly

combined to create two offspring $c^i_{1,2}$ by

$$c^i_{1,2} = 0.5 \cdot (1 \pm P(u)) \cdot p^i_1 + 0.5 \cdot (1 \mp P(u)) \cdot p^i_2 \tag{4.13}$$

$P(u)$ is the probability density function depending on a random number $u \in [0; 1]$:

$$P(u) = \begin{cases} (2u)^{\frac{1}{\nu+1}} & \text{if } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\nu+1}} & \text{otherwise} \end{cases} \tag{4.14}$$

High values for the distribution index ν provide a higher probability for creating individuals near their parents.
The multi-parent crossover operator PCX is a modified version of UNDX and is introduced by Deb [42]. μ individuals are selected as parents and the mean vector $\vec{g}$ is calculated of these individuals. The direction vectors are determined by $\vec{d}^{(p)} = \vec{x}^{(p)} - \vec{g}$. The offspring are created by a biased linear combination of three parents and are positioned around one of these parents $\vec{x}^p$:

$$\vec{y} = \vec{x}^p + w_1 |\vec{d}^{(p)}| + \sum_{i=1, i \neq p}^{\mu} w_2 \bar{D} \vec{e}^{(i)}, \tag{4.15}$$

where $w_{1,2}$ are zero-mean normally distributed variables. $\bar{D}$ is the average of the perpendicular distances D_i which are computed from each of the other $(\mu - 1)$ parents to the line $\vec{d}^{(p)}$. The goal of this operator is that offspring are created around each parent and the probability to remain close to the parents is higher.

4.4.1.3 Multi-parent Recombinations

Diverse recombination operators were extended to multi-parent recombination mechanism (e.g. see [53], [157]). The recombination mechanisms imitate the natural reproduction processes that are either asexual or sexual. In the case of asexual recombination, only one parent is used to create an offspring. The main disadvantage of this asexual reproduction is the lack of genetic material exchange from another parent. Biologically sophisticated individuals are characterized by sexual reproduction, therefore the reproduction of two parents to create offspring has become the state-of-the-art variant in ES history.
Bäck and Schwefel started with the recombination of information from more than two parents [12] in ES. In the empirical work of Eiben and Bäck, the impact of different multi-parent recombination strategies (from two parents up to sixteen) on the performance of a float-point genotype ES in the case of seven

test problems [52] is empirically examined. The aim of these experiments is the investigation of the question: Does increasing the number of parents lead to an improvement of the EA performance? In most cases, a significant performance increase is observed in combination with an increase in the parent number, although the algorithm performance depends on the recombination type and the optimization problem.
In the previous work of Eiben [53], genetic algorithms with multi-parent recombinations - generalized uniform and n-point crossover - have been extensively tested on different problem types: In most cases, more than two parents result in an increase of the GA performance, but the optimal number of parents varies greatly within the test runs. Otherwise, for some optimization problems the parent number has no influence on the GA performance. Furthermore, these extensive experiments reveal that the largest performance improvement is achieved by an increase of 2 to 3 parents. Eiben also proposes geometrical considerations of the parent number: A larger number of parents do not provide information about the same solution and consequently create offspring which are between but far away from their parents.

4.4.2 Recombination Operators Used in VONSEA

The recombination operators presented in the following are developed according to the model of nature, adjusted to the problem of peptide optimization and based on the conclusions from the empirical studies of the state-of-the-art recombinations. The following operators are classified into three categories: 'deterministic dynamic', 'position-specific' and 'Gaussian-distributed'. These operators determine either the number of recombination points or the position for recombination. All these recombination operators are used as multi-parent mechanisms. The default number of parents for recombination is three according to the work of Eiben [53] as described above. Nevertheless, the impact of the parent number on algorithm performance is the subject of the experiments on the different dimensional biochemical optimization problems that are presented in this thesis.

4.4.2.1 Deterministic Dynamic Recombination Operators

Two different deterministic dynamic recombination operators are optionally applied in VONSEA. These operators are n-point recombination operators,

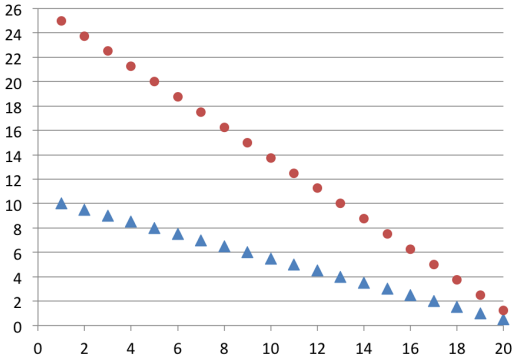

Fig. 4.3: LinDeRP: Number of recombination points for $l = 20$, $l = 50$ and $T = 20$

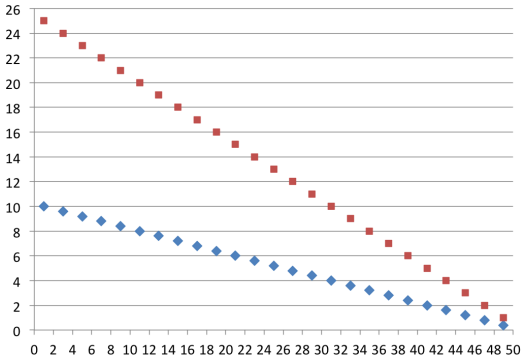

Fig. 4.4: LinDeRP: Number of recombination points for $l = 20$, $l = 50$ and $T = 50$

where the number n of recombinations points is determined by deterministic decreasing functions. The motivation of these operators is a high explorative search behavior in the early generations and a highly motifs-maintaining recombination in later generations. The velocity of the descent specified by a decreasing function determines the velocity of the transition from the explorative search to the motifs-maintaining recombination and therefore local search. These recombination operators have been introduced in [131].

The Linear Decreasing ReProduction operator (LiDeRP) varies the number of recombination points over the generations via a linearly decreasing function:

$$x(t) = \frac{l}{2} - \frac{l/2}{T} \cdot (t-1), \tag{4.16}$$

which depends on the length of the individual l, the total number of the GA generations T and the actual generation number t. The number of recombination points in the first generation ($t = 1$) is $l/2$ and decreases linearly until one recombination point in the last generation ($t = T$). The recombination points themselves are determined randomly, but it is not excluded that recombination points are determined more than once. The magnitude of the preserved motifs especially in the later generations increases. Fig. 4.3 and Fig. 4.4 exemplarily depict the number of recombination points for a peptide length of 20, 50 and a total number of generations of 20 or 50 respectively.

The Exponential Decreasing ReProduction operator (ExpoDeRP) determines the number of recombination points over the generations by an exponentially decreasing function:

$$x_R(t) = 2 + (0.2 \cdot l - 1) \cdot 2^{\frac{-l/2}{T} \cdot (t-1)} \tag{4.17}$$

The number of recombination points in the first generation ($t = 1$) is $(0.2 \cdot l + 1)$ and decreases exponentially until two recombination points in the last generation ($t = T$). The recombination points themselves are determined randomly,

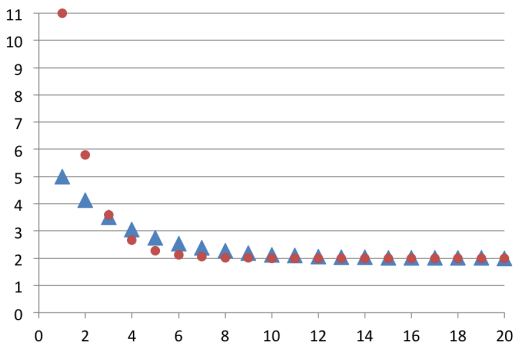

Fig. 4.5: ExpoDeRP: Number of recombination points for $l = 20$, $l = 50$ and $T = 20$

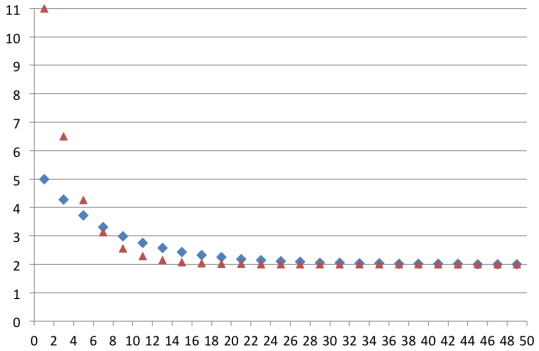

Fig. 4.6: ExpoDeRP: Number of recombination points for $l = 20$, $l = 50$ and $T = 50$

but it is not excluded that recombination points are determined more than once. Fig. 4.5 and Fig. 4.6 exemplarily depict the number of recombination points for a peptide length of 20, 50 and a total number of generations of 20 or 50 respectively.

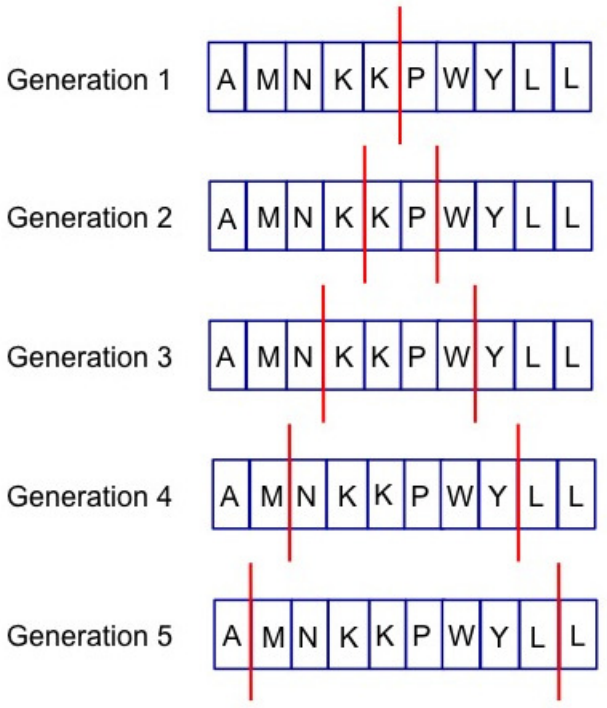

Fig. 4.7: Positions of recombination points by 2-point-edges exemplarily depicted on a peptide of the length 10 over 5 generations.

4.4.2.2 Position-specific Recombination Operator

One position-specific recombination operator is used in VONSEA. This operator is a dynamic 2-point recombination operator, where the recombination points move linearly from the middle of a peptide to the ends of the peptide sequence. The motivation for this operator is an increasing motifs-maintenance of the genetic material over the generations and a variation of the C- and N-termini of the peptides in later generations. The ends of a peptide have a specific influence on the properties of peptides (e.g. [18]). In the first generation the peptides are divided into two parts (see Fig. 4.7). In the following generations the genetic material of the peptide ends are exchanged, where the number of amino acids at the ends is decreasing over the generations. In the last generation, only the N-terminus[4] and C-terminus[5] are exchanged. Therefore, the recombination in the last generation is interpretable as mutation. The recombination points are determined by the following two functions:

$$p_1(t) = \frac{l}{2} - \frac{l/2}{T} \cdot (t-1) \tag{4.18}$$

$$p_2(t) = \frac{l}{2} + \frac{l/2}{T} \cdot (t-1). \tag{4.19}$$

This operator is further termed '2-point-edges' and has been introduced in [131].

4.4.2.3 Random Recombination Operator

A Gaussian-distributed recombination operator is used in VONSEA. This operator is a n-point recombination operator, where the number n of recombination points is varied according to a Gaussian distribution. This recombination method is an imitation of the natural recombination procedure. The number n of recombination points is determined by the integer result of the product of the individual length and a Gaussian distributed random number. The actuarial expectation as the most frequent number has the highest probability to be selected as the number of recombination points. Therefore, the parameters of the recombination operator are the parameters of the Gaussian distribution, the actuarial expectation ρ (the most frequent number for n) and the standard

[4]The N-terminus is determined as the start of a peptide that is terminated by the amino acid with a free amine group (NH_2). The N-terminus of a peptide is important for the determination of its half-life.

[5]The C-terminus is the end of a peptide that is terminated by a free carboxyl group ($-COOH$) and it contains retention signals.

deviation σ. The default values are set to $\rho = 2$ and $\sigma = 2.5$. Obviously, only positive values $n \geq 0$ are permitted, therefore negative values are multiplied by -1. This recombination operator has been introduced in [130].

4.5 Mutation Operators in VONSEA

The mutation operator in evolutionary algorithms is a reproduction operator which is primarily applied for diversity preservation regarding the genetic material. It operates on single solutions of a population and modifies them independently of the remaining members of the population in contrast to the recombination operator. The important role of a mutation operator is the overall search efficiency by avoiding premature convergence. Mutation strategies are mainly evolved in the field of ES, but several mutation procedures have also been proposed for GAs according to the individual encoding. More effort and research have been investigated in the evolution of the variation operator 'mutation' than for 'recombination' to improve the performance of a GA. The first part of the following section describes the mutation operators evolved in the area of GAs. The following part describes a special category of mutation operators evolved in the area of ES which make use of dynamic and self-adaptive parameters to find the optimal mutation rate for improving the algorithm performance.

4.5.1 Traditional Mutation Operators

Several mutation operators have been introduced in the field of GAs depending on the GA encoding. The mostly used operator in binary-coded GAs is the bitwise mutation operator. A bit is flipped with a specific probability p_m. The choice of $p_m = \frac{1}{l}$ achieved the best performance in comparative studies as demonstrated by Schaffer et al. [137], where l denotes the number of total bits. Different operators have been proposed for real-coded GAs: The most popular operators are the parameter-based mutation operator [38], the Gaussian [141] and the polynomial mutation operator [39].
Parameter-based mutation uses a polynomial probability distribution and a user-defined parameter ν to create a new solution p' in the vicinity of a solution $p \in [x_l; x_u]$, where x_l and x_u are the lower and upper bounds of a variable. The creation of a new solution is based on this probability distribution enclosing

the variable values by an upper and lower boundary. A mutated solution p' of a parent solution p is created with a random number $u \in [0;1]$ by

$$p' = \begin{cases} p + \delta_L(p - x_i^{(L)}), & \text{for } u \leq 0.5 \\ p + \delta_R(x_i^{(U)} - p), & \text{for } u > 0.5, \end{cases} \tag{4.20}$$

where the two parameters δ_L and δ_R are calculated by

$$\delta_L = (2u)^{\frac{1}{\eta_m+1}} - 1 \text{ for } u \leq 0.5, \tag{4.21}$$

$$\delta_R = 1 - [2(1-u)]^{\frac{1}{\eta_m+1}} \text{ for } u > 0.5 \tag{4.22}$$

The parameter η_m is the distribution index for the mutation that takes any non-negative value and determines the peakedness of the distribution. A value $\eta_m \in [20;100]$ revealed to be adequate in most optimization problems. For small values of η_m, new solutions are produced far away from their parents. Higher values for η_m result in higher probabilities of new solutions within the vicinity of the parents. The spread factor δ is calculated via a probability distribution. A problem of the original polynomial mutation is that the mutation becomes useless in the case of very small spread factors δ. Algorithms using polynomial mutation often trap in local optima, especially in the case of multi-modal problems. The polynomial mutation was originally introduced by Deb [37], used in NSGA [143] and in the early version of NSGA-II. It was later improved by Deb and Tiwari [46], and Carvalho et al. proposed an adaptive version of this operator in NSGA-II [26]. The improved polynomial mutation [46] differs from the original one in the choice of δ. This modified version allows big jumps within the search space and therefore does not stick in local optima. But the big jump potentially results in unsmooth approximations to the Pareto front. The adaptive mutation of Carvalho [26] uses information about the diversity of the population through the component crowding distance in NSGA-II to control the strength of the mutation. Thereto, the parameter η_m is changed adaptively using information about the greatest and lowest distance value of $i_{distance}$ (calculated by the component crowding distance for each solution i of the population with $i_{distance} \neq \infty$) and about the current stage of the evolutionary process. This provides high mutation rates in early generations of the genetic process and the rates are reduced during the process.

The classical mutation method in the field of ES is the Gaussian mutation which was described by Bäck and Schwefel [12]. Gaussian mutation has also been used in real-coded GAs [36]. According to [12], a new solution is created by the addition of a scaled Gaussian normally distributed random number to

the previous parameter values. The motivation for this mutation is to avoid the sticking in a vicinity of low-qualified solutions. For an improvement of new solutions, larger mutation probabilities are required, but occur only occasionally. This lead to the idea of using continuous variables and changing them with values determined by Gaussian distribution. A solution is presented as a pair of real-valued vectors (x_i, σ_i), where x_i is the ith variable and σ_i is the associated strategy parameter. A new solution (x_i', σ_i') from a parent (x_i, σ_i) is determined by

$$\sigma_i'(j) = \sigma_i(j) \cdot exp(\tau \mathbb{N}(0,1) + \tau' \mathbb{N}_i(0,1)) \tag{4.23}$$

$$x_i'(j) = x_i(j) + \sigma_i'(j) \cdot \mathbb{N}_j(0,1), \tag{4.24}$$

where $\mathbb{N}(0,1)$ is a normally distributed random number, $\mathbb{N}_j(0,1)$ is a different random number for each j and $x_i'(j)$, $x_i(j)$, $\sigma_i'(j)$, $\sigma_i(j)$ denote the jth component of the vectors x_i', x_i, σ_i', σ_i. According to Schwefel, the parameters τ and τ' are typically set to $\frac{1}{\sqrt{2n}}$ and $\frac{1}{\sqrt{2\sqrt{n}}}$ [8].

It is a well-known fact that GA performance is influenced by the mutation rate (p_m) and recombination rate (p_r) as well as the interaction of these two basic variation operators. The mutation and recombination rates are usually kept static. Optimal setting for these rates possibly improve GA performance, but empirical tuning is very time consuming and practically impossible in examining all combinations of p_m and p_r systematically. Furthermore, several studies have shown that the variation of mutation probability is preferable to a constant mutation rate ([5], [6], [11]). A solution for this problem is the introduction of dynamically changing parameters in the mutation. Different dynamic and self-adaptive parameters have also been proposed in the field of EA. (e.g. [56])
These mutation operators with dynamically changing parameter values are classified into three types [9]:

- deterministic dynamic (the parameters are varied by deterministic functions usually depending on the generation number)
- (dynamic) adaptive
- (dynamic) self-adaptive.

The mutation operators associated with these categories are presented in the following.

4.5.1.1 Deterministic Dynamic Mutation Operators

The early dynamic mutation operator was introduced by Fogarty [63] in the early 1960s. The mutation rate decreases as the generation number increases. The main idea of this operator is that high mutation rates in early generations favor exploration and lower mutation rates in later generation favor exploitation. Mutation probabilities are determined via an exponentially decreasing function depending on the actual generation number t:

$$p_m(t) = \frac{1}{240} + \frac{0.11375}{2^t}. \tag{4.25}$$

Alternatively, Fogarty proposed a mutation operator for binary representations by changing the mutation rates per bit: For $j = 1, ..., n_b$ (with n_b is being the least significant bit):

$$p_m(j) = \frac{0.3528}{2^{j-1}} \tag{4.26}$$

Bäck and Schütz proposed a mutation operator that determines the mutation probability via a deterministic decreasing function [11]. The motivation for the evolution of this mutation operator is that higher mutation probabilities in early generations of a GA lead to a good exploration. Lower mutation probabilities in later generations provide a good exploitation in the local area of the landscape. The mutation rates are calculated via:

$$p_{BS}(t) = (2 + \frac{l-2}{T-1}t)^{-1}, \tag{4.27}$$

where T is the maximal number of generations, l is the length of the individual and t the actual generation number. The mutation rates are bounded by $(0; \frac{1}{2}]$, the initial generation has a mutation rate of $\frac{1}{2}$.

4.5.1.2 Dynamic Adaptive Mutation Operators

The early adaptive mutation operator is Rechenberg's '$\frac{1}{5}$ success rule' [127]. The basic idea is to control the parameter values by a feedback from the performance of the search process. The mutation strength σ is increased if the ratio of successful candidate solutions is greater than $\frac{1}{5}$. σ is decreased if the ratio is less than $\frac{1}{5}$. In general, $\sigma \in (0, 1)$.
Thierens introduced the dynamic adaptive mutation operator for binary strings termed 'Thierens' constant gain adaptive mutation scheme' [153]. The mutation scheme tries three different mutation rates on a current individual. The

comparison of the fitness values of the three offspring gives a rough hint whether the current mutation rate should be increased or decreased. The modification of the current mutation rate is carried out proportionally by multiplying or dividing the current rate with the constant learning factor α. During the evaluation, a factor ω called exploration factor is used. Usually $\omega > \alpha > 1$ to avoid oscillations of the mutation rates. Formally $M(x, p_m) \longrightarrow (x^*, p_m^*)$ symbolizing that the individual x with mutation rate p_m generates the offspring x^* with the new mutation rate p_m^*. The mutation scheme of Thierens:

1. Mutate the current individual (x, p_m):
   $M(x, p_m/\omega) \longrightarrow (x_1, p_m/\alpha)$
   $M(x, p_m) \longrightarrow (x_2, p_m)$
   $M(x, p_m \cdot \omega) \longrightarrow (x_3, p_m \cdot \alpha)$

2. Select the fittest individual of
   $\{(x, p_m), (x_1, p_m/\alpha), (x_2, p_m), (x_3, \alpha \cdot p_m)\}$

Thierens advised appropriate values with $\alpha = 1.1$ and $\omega = 1.5$. Furthermore, Thierens proposed a variant of the constant gain scheme [153]: The 'Declining adaptive mutation scheme' allows a more aggressive step size within the mutation probabilities than the constant gain method, but it suppresses the oscillating behavior caused by the learning factor α. The current mutation probability of a current individual is decreased by a small factor termed declination factor γ. The procedure of Thierens' declining adaptive scheme is defined as:

1. Mutate the current individual (x, p_m):
   $M(x, \omega \cdot p_m) \longrightarrow (x_1, \alpha \cdot p_m)$
   $M(x, p_m) \longrightarrow (x_2, p_m)$
   $M(x, p_m \cdot \omega) \longrightarrow (x_3, p_m \cdot \alpha)$

2. Decrease the mutation probability of the parent $(x, p_m) \longrightarrow (x, \gamma \cdot p_m)$

3. Select the fittest individual of
   $\{(x, \gamma \cdot p_m), (x_1, \alpha \cdot p_m), (x_2, p_m), (x_3, \alpha \cdot p_m)\}$

Appropriate choices are published as $\omega = \alpha = 2.0$ and $0.9 \leq \gamma < 1$. The mutation schemes of Thierens are not transferable to a MOEA without adaption in step 2 or 3 respectively, as a strategy to determine the fittest individual in the multi-objective sense is challenging. In general, two alternatives are possible: The fittest individual in the multi-objective sense is determined by a Pareto front ranking or alternatively by an appropriate indicator.

The ACV-indicator ACV_{scaled} as introduced in section 5.2.2 is used as such an indicator in an adapted version of Thierens' constant gain adaptive mutation scheme. The individual with the lowest ACV_{scaled} is the fittest in the multi-objective sense. The alternative of the Pareto front ranking is not in the focus as the low number of individuals has a high potential to be indifferent or multiple solutions are ranked into the first front. As a consequence, the determination of the succeeding mutation rate has to be done by chance.

4.5.1.3 Self-adaptive Mutation Operator

Bäck and Schütz introduced the self-adaptive mutation operator [11]. The motivation for the evolution of this mutation operator is the idea that individuals with good parameter choices receive an evolutionary advantage and the individuals themselves will proliferate in the population. This self-adaptive operator was originally designed for binary strings. The mutation probabilities are calculated via the following function:

$$p_m(t+1) = (1 + \frac{1 - p_m(t)}{p_m(t)} \cdot e^{-\gamma N(0,1)})^{-1}, \tag{4.28}$$

where $N(0,1)$ is a normally distributed random number and the learning rate γ controls the adaption steps of the mutation rate. A traditional choice for the learning rate is $\gamma = 0.22$. This operator has desirable features: $p_m(t) \in (0,1) \Rightarrow p_m(t+1) \in (0,1)$ and small changes between the probabilities of consecutive mutation rates are more likely than large ones.

4.5.2 Mutation Operators used in VONSEA

The adapted variant of Thierens' constant gain adaptive mutation scheme as well as the self-adaptive mutation operator of Bäck and Schütz, as presented in the last section, are used in VONSEA. Further mutation operators are evolved for VONSEA to solve molecular optimization problems and belong to the types 'deterministic dynamic' and 'random' mutations. The random mutation operators are evolved with the aim of imitating the natural mutation procedure. The general idea of deterministic dynamic mutation operators is: High mutation rates in early generations support the explorative search and therefore allow the discovery of new regions of the fitness landscape. Lower mutation rates in later generations support the exploitive search, which allows

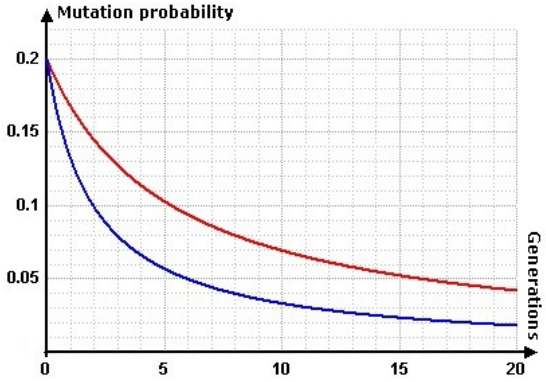

Fig. 4.8: Mutation rates of p_{adapBS} with $a = 5$, $l = 20$, $l = 50$ and $T = 20$

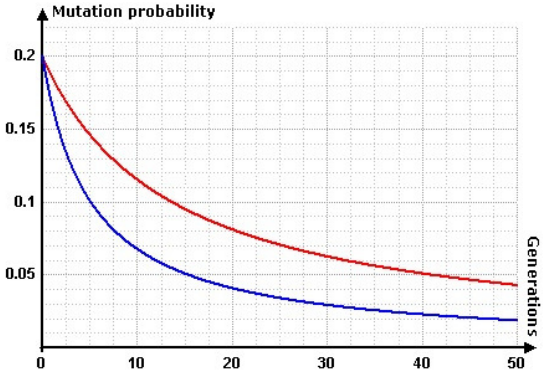

Fig. 4.9: Mutation rates of p_{adapBS} with $a = 5$, $l = 20$, $l = 50$ and $T = 50$

the convergence to other optima in the vicinity. The start mutation rates are set to $p_0 = 0.2$ exceeding the random mutation operators.

4.5.2.1 Deterministic Dynamic Mutation

Three different deterministic dynamic mutation operators are used in VONSEA. They have in common that mutation probability is determined by a decreasing function depending on the actual and total number of generations. The difference between these decreasing functions is the level of the mutation rate decrease:
The deterministic dynamic mutation operator of Bäck and Schütz is implemented in VONSEA in an adapted version: The start mutation rate is reduced as it is not used in a mutation-only GA. According to [117], sufficiently low mutation rates are preferred in GAs with a recombination operator. Apart from this, a high mutation rate in the early generations results in an inappropriately high destruction of the sequence structure and the offspring are highly different in appearance compared to their parents. As a consequence, a high mutation rate increases the probability that an optimal partial structure is lost by the succeeding population. The mutation probabilities are determined by the function

$$p_{adaptBS} = (a + \frac{l-2}{T-1}(t-1))^{-1}, \tag{4.29}$$

where T is the total number of generations and t is the actual generation number. The usual choice of the parameter is $a = 5$. Mutation rates here are bounded by $(0; \frac{1}{4}]$. Fig. 4.8 and Fig. 4.9 depict the mutation probabilities of the adapted deterministic dynamic operator for different maximal generation numbers and solution lengths as functions. This adapted version has been

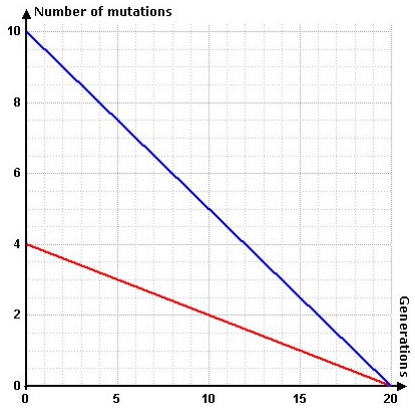

Fig. 4.10: LinDeMut: Number of mutations for $l = 20$, $l = 50$ and $T = 20$

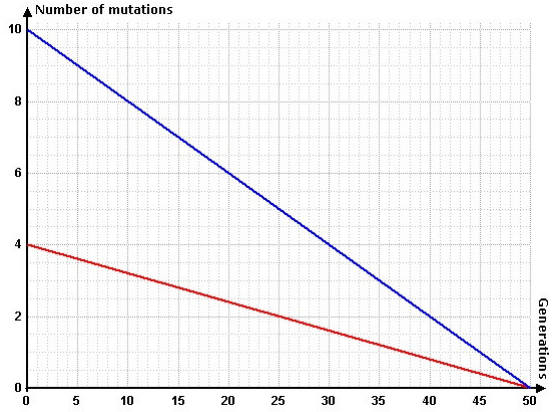

Fig. 4.11: LinDeMut: Number of mutations for $l = 20$, $l = 50$ and $T = 50$

introduced in [132]. Herein it is benchmarked for a three-dimensional biochemical minimization problem, and the customized NSGA-II with this mutation operator yields pretty good results with regard to early convergence and a good diversity within the solutions.

Based on these experiences, two further deterministic dynamic mutation operators are evolved according to this model and have been proposed in [131]. The first mutation operator is the mutation 'LiDeMut' which varies the number of mutations by a linearly decreasing function:

$$x_M(t) = \frac{l}{5} - \frac{l/5}{T} \cdot (t-1), \tag{4.30}$$

where T and t are once again the total or the actual generation number. LiDeMut is developed taking account of specific requirements: The initial mutation rate is 0.2 and the number of mutations is 1 in the latest generation independent of the individual length. Fig. 4.10 and Fig. 4.11 depict the numbers of mutations of LinDeMut for different maximal generation numbers and solution lengths as functions.

The second mutation operator is the mutation 'QuadDeMut', which determines the number of mutations by a quadratically decreasing function:

$$x_M(t) = \frac{1 - l/5}{T^2} \cdot (t-1)^2 + \frac{l}{5}, \tag{4.31}$$

where T and t are the total or the actual generation number. QuadDeMut is evolved under the same requirements as LiDeMut. Furthermore, the mutation rates have to be changed slightly within the first few generations, meaning that the function that describe the number of mutation changes has to be a shrunken quadratic function for each individual length. Fig. 4.12 and Fig. 4.13 depict the numbers of mutations of QuadDeMut for different maximal generation numbers and solution lengths as functions.

Fig. 4.12: QuadDeMut: Number of mutations for $l = 20$, $l = 50$ and $T = 20$

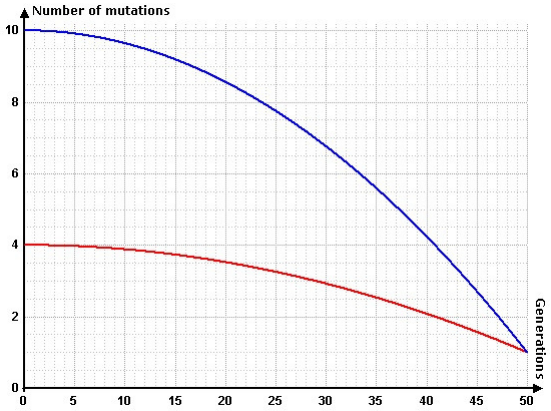

Fig. 4.13: QuadDeMut: Number of mutations for $l = 20$, $l = 50$ and $T = 20$

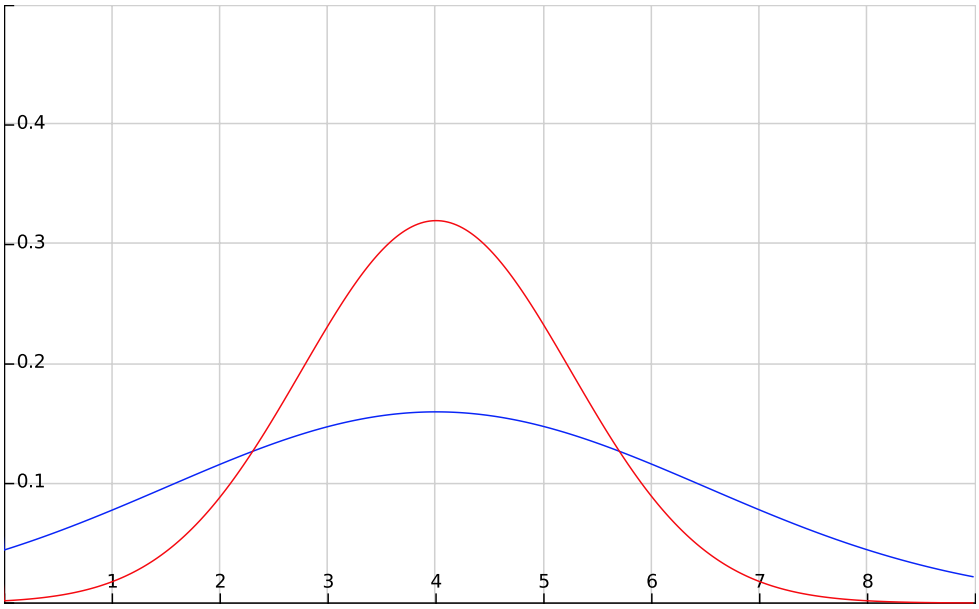

Fig. 4.14: Shapes of the probability density functions $N(4, 1.25)$ (red) and $N(4, 2.5)$ (blue).

4.5.2.2 Random Mutation

Two mutation operators are evolved in an highly intuitive way according to the model of nature. For both mutation schemes, it holds that the number of mutations is determined randomly by a Gaussian distributed random number. The parameters of these mutation operators are the parameters of the Gaussian distribution, the actuarial expectation μ and the standard deviation σ. The default values are $\sigma = 1.25$ and $\mu = 4$, denoted as $N(4, 1.25)$. The shapes of the probability density functions $N(4, 1.25)$ and $N(4, 2.5)$ are depicted in Fig. 4.14. These default values are set in the way that the most frequent

number ($\mu = 4$) corresponds to the start mutation rates of the other mutation operators and a small shape of the density function ($\sigma = 1.25$) is preferred to provide only a low range of mutation variations. The number of mutations is the integer result of the product of the individual length and the Gaussian distributed random number. Once more, mutation numbers have to be non-negative and are otherwise multiplied with -1.

The difference between the mutation operator termed 'Random' to its alterna-

	2nd position				
1st position	U	C	A	G	**3rd position**
U	Phe F	Ser S	Tyr Y	Cys C	U
	Phe F	Ser S	Tyr Y	Cys C	C
	Leu L	Ser S	–	–	A
	Leu L	Ser S	–	Trp W	G
C	Leu L	Pro P	His H	Arg R	U
	Leu L	Pro P	His H	Arg R	C
	Leu L	Pro P	Gln Q	Arg R	A
	Leu L	Pro P	Gln Q	Arg R	G
A	Ile I	Thr T	Asn N	Ser S	U
	Ile I	Thr T	Asn N	Ser S	C
	Ile I	Thr T	Lys K	Arg R	A
	Met M	Thr T	Lys K	Arg R	G
G	Val V	Ala A	Asp D	Gly G	U
	Val V	Ala A	Asp D	Gly G	C
	Val V	Ala A	Glu E	Gly G	A
	Val V	Ala A	Glu E	Gly G	G

Table 4.2: Table of the code sun: coding of the amino acids by base triples.

tive denoted as 'AAweighted' is that AAweighted is an imitation of nature with regard to the selection of the characters symbolizing the 20 canonical amino acids. In AAweighted, each of these 20 characters (canonical amino acids) has its specific frequency to be mutated to (according to their natural incidence). Table 4.2 shows the table of the circle of the decoded 20 amino acids using the messenger Ribonucleic Acid (mRNA) codon that is the basis for the frequency.

4.6 Selection Strategies of VONSEA

The major task of the selection procedure within an evolutionary strategy is to scan the search space and to guide the search in the direction of high quality solutions. In general, the selection in EA has two tasks: The selection of the parent solutions for reproduction and the determination of the members of the succeeding generation. The choice of the selection strategy has a considerable influence on algorithm performance and particularly on convergence velocity [49]. There are no fixed rules regarding an appropriate selection strategy for all optimization problems. A selection strategy that prefers high quality solutions and discards low ones does not necessarily lead to a good algorithm performance as low quality solutions have in some cases useful genetic material that is able to produce high quality solutions in later generations. In general, a good selection strategy is characterized by a good balance between exploration and exploitation for a well-spread solution set over the Pareto front. Bäck shows in his study of selective mechanisms that the selection process controls the balance of exploration and exploitation by varying the guidance of the search process in the direction of the fittest individuals [7]. A selection process that is more stringent in the direction of the fitter solutions is oriented towards exploitation, whereas a less stringent selection is oriented towards exploration.

4.6.1 Traditional Selection Strategies

Several selection strategies have been evolved for EA with the main focus on the phenotype or genetic material that is to be passed on the gene pool of the succeeding generation. Bäck et al. proposed a classification of selection strategies with respect to different criteria [7]:

1. Classifications according to the selection probabilities of the solutions

    - Dynamic or static selections
      Selection strategies are characterized under the terms of dynamic or static selection probabilities. A selection is classified as dynamic if the selection probabilities of the individuals depend on their actual fitness values and are different in each generation. In the case that the fitness values depend only on the ranking of the fitness values and in addition these fitness values are fixed for all generations, the selection is characterized as static.

- Extinctive or preservative selections
  A selection is termed extinctive if some individuals are excluded from reproduction. These individuals have a probability of zero to be selected. Thus, a selection is termed preservative if all solutions have a non-zero probability to be selected for reproduction.
  Extinctive selections are further classified into left and right extinctive selection: An extinctive selection is termed right if the worst individuals achieve a selection probability of zero. In the case that the high quality individuals achieve a selection probability of zero and are not selected for reproduction to avoid premature convergence, the name left extinctive selection is used.

2. Classification according to the life or reproduction time of the solutions

   - Elitist or pure selection
     Within a pure selection, each individual has a life time of only one generation independent of its fitness values. If all individuals - parents and offspring - undergo the selection with the result that high quality individuals achieve an unlimited life time, this is referred to as elitist selection.

   - Generational or steady-state selection
     In the case of generational selection, the parent set is fixed until all members of the succeeding generation have been produced. A special variant of elitist selection is steady-state selection; offspring replace parents only if they are of higher quality. Therefore, the parents set changes for each reproduction step.

A formal description of these classifications referring to (μ, λ) selections in ES are presented by Bäck and Hoffmeister [10].
In this thesis, a further characterization of selection strategies is proposed: Selection strategies are characterized according to their fundamental issues in an evolutionary process. Three objectives are defined that are not excluding:

- Diversity of the genetic material
  This objective refers to the production of an maximally high diversity of the genetic material within the succeeding generation.

- Maximization of the solutions spread
  This objective refers to the detection of high quality individuals with maximally wide spread among themselves.

- High fitness-directional guidance
  This objective refers to the strong guidance of the selection process in direction of the highest quality individuals.

The commonly-used selection strategies within GAs are described and characterized according to the classifications and objectives presented above:

4.6.1.1 Truncation Selection Strategies

The truncation method is the most natural selection as well as the most useless in GAs as it does not give any chance to low quality solutions to be selected. Therefore, potentially high quality genetic material in low quality solutions is lost. The procedure starts by sorting the individuals according to their fitness. The pre-defined parameter 'truncation threshold' T is used to select only the best T individuals. A common choice for T and therefore for the proportion of individuals with the opportunity of producing offspring are values in the range of 50% to 10%. Truncation selection is characterized as a right extinctive - since it excludes low quality solutions from the selection by the threshold parameter - and dynamic strategy, as the sorting of the individuals depends on their actual fitness values. High fitness-directional guidance is the main selection objective of this strategy.

4.6.1.2 Fitness-proportionate Selection

This category of selection strategies includes stochastic methods. Fitness- proportionate selection was first described by Holland [84]. The main idea of these methods is that every individual has a non-zero chance to be selected: High quality solutions have a higher probability to be selected than lower ones. The selection probability of every individual is therefore proportional to its fitness. Furthermore, a scaling of the fitness values is usually necessary to map the values into a suitable range. This category of selection strategy is characterized as static and preservative as each individual has a non-zero probability to be selected [152]. Diversity of the genetic material and solutions spread are the main selection objective of these methods.

The most common strategy of this category is Roulette Wheel Selection (RWS). Each individual is assigned a segment on an imaginary roulette wheel. The size of the segments is proportional to the individuals' fitness. The selection procedure - an individual is selected by spanning the wheel and a pointer

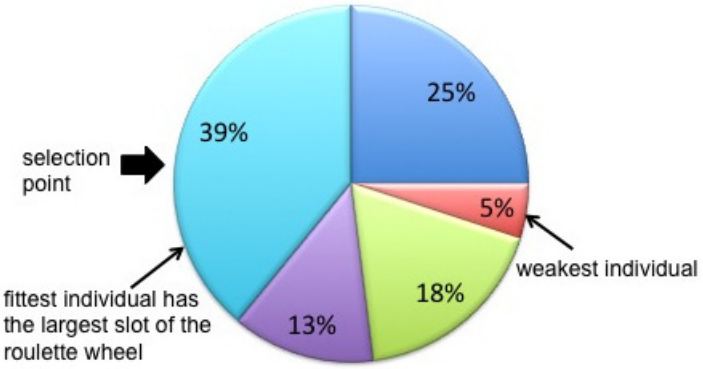

Fig. 4.15: Illustration of RWS for a five-member population. Each individual is assigned a segment whose size is proportional to its fitness. The percentages denote the selection probabilities.

determines the segment as well as the associated individual - is described as follows:

1. Determination of the total fitness F as the sum of all solutions.
2. A random number n is chosen in the interval $[0; F]$.
3. The individual with a fitness greater or equal n is returned.

Steps $1. - 3.$ repeat until the desired number of individuals is selected. The probability of the j-th individual to be selected is given by $P(f_j) = \frac{f_j}{\sum_{i=1}^{N} f_i}$, where N is the number of individuals in a generation and f_i the fitness of the individual x_i.
The selection process is guided in the direction of high quality individuals. The main disadvantage of RWS is that there is no guarantee for the best individual to be selected.

Stochastic Universal Sampling (SUS) is the most widely used strategy in GA literature nowadays and was originally developed by Baker [14]. SUS is motivated by overcoming the main disadvantage of RWS. It is a single-phase sampling based on the RWS approach: Individuals are assigned to segments on the roulette wheel that are equal in size to their fitness. However, SUS differs in the number of selection pointers: Instead of a single pointer, K pointers are equally spaced around the wheel, where K is the number of selections required. The selection procedure is described as follows:

1. Determination of a single number $ptr \in [1; F/K]$ with the total fitness F.

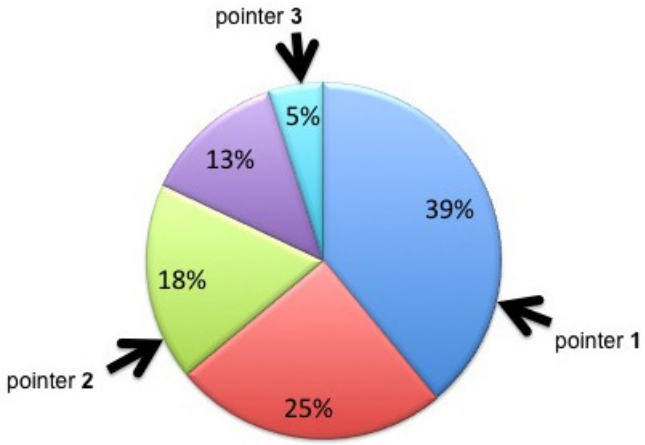

Fig. 4.16: Illustration of SUS for a five-member population. A spinner with three evenly spaced pointers is spun once to obtain three individuals.

2. Individuals are chosen by generating K pointers on the wheel, starting with pointer ptr and the remaining are spaced by 1: $[ptr, ptr+1, ..., ptr+N-1]$.

As the individuals are selected according to their position in the population, SUS has zero bias[6]. Furthermore, the number of copies of a solution is equal to the number of pointers that are pointed on the associated segment. Therefore, an individual x_i is selected between $N_{i,min} = [N \cdot f_i/F]$ and $N_{i,max} = [N \cdot f_i/F]$ times. On this account, SUS provides minimum spread[7].
These two features - zero bias and minimum spread - are preferred features for a selection method and make SUS one of the best established selection strategy in the area of EA.

4.6.1.3 Rank-based Selection

Rank-based selection (RBS) is a strategy that assigns a selection probability to each individual on the basis of its fitness value relative to the entire population. Rank-based selection was first suggested by Baker [13] and was evolved in order to overcome the scaling problem of the direct fitness based strategies. RBS uses a mapping function for this probability assignment that is either a linear or a non-linear function. The selection performance strongly depends on this mapping function. The selection procedure starts by sorting the population

[6] Absolute difference between an individual's normalized fitness and its expected probability of reproduction. [14]

[7] The range of possible values for the number of offspring assigned to an individual.

according to the individuals fitness values. Then, selection probabilities are assigned to each individual via the linear formula in the case of linear rank-based selection:

$$rank(pos) = 2 - SP + \left(2(SP-1) \cdot \frac{pos-1}{N-1}\right) \tag{4.32}$$

where *pos* is the position of the individual in the sorted list of the population and SP is the parameter selection pressure of RBS. SP controls the bias and is chosen as $2 \geq SP \geq 1$. The sampling rate of the best individual is SP and $(2 - \frac{1}{2}SP)$ for the worst one. RBS requires a further selection method such as RWS to select the individuals. The selection probability of the solutions depends in the their position in the sorted list. As the n-th individual in one generation has the same selection probability as the n-th individual in the succeeding generation, rank-based selection is categorized as static [74].
The advantage of RBS is the uniform scaling across the population: The selection probabilities are assigned regardless of the concrete fitness values. This prevents premature convergence to 'super'-individuals as the fittest individual always achieves the same selection probability. Bäck et al. showed that RBS is more robust than other methods [10]. A disadvantage of this strategy is the computational complexity caused by the necessary sorting of the population.

4.6.1.4 Tournament Selection

Tournament selection is the most widely used selection strategy due to its efficiency: No fitness scaling or sorting is necessary and it is simple to implement [179]. Its computational complexity is $O(kN)$, where k is the number of random individuals from the population and N the population size. In the case of a tournament size of $k = 2$, the terminology binary tournament selection is used [44]. A particular number k of individuals are selected from the population and set to competed with each other. The fittest one according to a specific criterion is selected. The main selection objectives of this strategy are the solutions spread and the fitness-directional guidance provided by controlling the selection pressure via the parameter 'tournament size'. Tournament selection is characterized as dynamic and right extinctive as the selection probability depends on the actual fitness values and the worst individual has a zero probability to be selected in the succeeding generation.
The particular number k is termed tournament size and is challenging: The convergence rates of an EA are determined by selection pressure:[8] the hig-

[8]Selection pressure is the degree to which the high quality solutions are favored. [14]

her the selection pressure, the higher are the convergence rates. Otherwise, selection pressure is influenced by tournament size [176]: Larger tournament sizes ensure higher selection pressure. Blickle and Thiele proved in their work that tournament selection has the smallest loss of diversity[9] and the highest selection variance[10] for the same selection intensity[11] compared to truncation selection [19]. Furthermore, they proved that binary tournament selection and linear ranking have identical average behavior.
Zhong et al. compare the proportional roulette wheel with tournament selection with a tournament size of 6 on several general test functions [179]. Tournament selection reveals itself as more efficient in convergence. Razali et al. compare tournament selection, proportional roulette wheel and rank-based roulette wheel in solving the Traveling Salesman Problem [126]. Tournament selection outperforms both. Tournament selection and proportional roulette wheel are superior to rank-based roulette wheel in the case of small problems, but they sometimes become susceptible to premature convergence in the case that the problem size increases.
Due to these advantages, the popularity of tournament selection is growing rapidly and a number of variants has been developed. A short overview of these variants is presented by Xie et al. [176].

4.6.2 Selection Strategies of MOEA

The selection strategies presented in this section have been developed in recently introduced and established MOEAs. Their common feature is the selection objective: a good spread of the individuals over the Pareto front. These strategies provide different schemes of assigning higher selection probabilities to the individuals in less crowded areas of the objective space.

4.6.2.1 Indicator-based Selection

The selection strategies of this category make use of an indicator as a selection criterion. IBEA [182] determines fitness values for each individual based on

[9] The proportion of individuals of a population that is not selected during the selection phase [19]

[10] The expected variance of the fitness distribution of the population after applying a selection method to the normalized Gaussian distribution [19]

[11] The expected average fitness value of the population after applying a selection method to the normalized Gaussian distribution [19]

a binary quality indicator and selects the individuals for the mating pool by binary tournament selection. The fitness assignment is performed via three iteration steps as long as the population size is not exceeded:

1. The individual x^* with the smallest fitness value (least fit individual) is selected and deleted from the population P.
2. $P \longleftarrow P \backslash \{x^*\}$
3. The fitness values of the remaining individuals are updated by $F(X) = F(X) + e^{-I(x^*,x)/K}$ for all $X \in P$, where I is the binary quality indicator and K the fitness scaling factor.

The commonly used binary quality indicators with the aim of comparing the quality of two different Pareto optimal sets are the ϵ-indicator [182]

$$I_{\epsilon+}(A,B) = min_{\epsilon+}\{\forall x_2 \in B, \exists x_1 \in A : f_i(x_1) - \epsilon \leq f_i(x_2) \text{ for } i \in \{1,...,n\}\} \tag{4.33}$$

and the hypervolume indicator. The latter one presents the volume of the space that is dominated by the solution set A, but not by B:

$$I_{HD}(A,B) = \begin{cases} I_H(B) - I_H(A) \text{ if } \forall x_2 \in B,\ \exists x_1 \in A : x_1 \succ x_2 \\ I_H(A+B) - I_H(A) \text{ otherwise} \end{cases} \tag{4.34}$$

In general, several other dominance preserving indicators are potential indicators for the selection strategy. Binary tournament selection is further used to fill the temporary mating pool in IBEA. As the hypervolume is used as a measure for convergence and diversity (see 5.2.1), a good spread of high quality solutions is ensured.

Another indicator-based selection strategy is the selection based on dominated hypervolume within the steady-state SMS-EMOA [17]. This strategy is a steady-state selection and the selection criterion is based on the hypervolume measure $S(X)$ [54]. SMS-EMOA stores non-dominated and dominated solutions in a population of constant size. This classification is realized by the non-dominated sorting of NSGA-II. In each iteration, a new individual is produced. The selection decides if this new individual s_0 enters the population P. The hypervolume-based selection criterion ensures that no non-dominated individual is replaced by a dominated one, therefore the selection strategy is regarded as elitist. The selection procedure decides which individual is kept in the population by deleting the least fittest individual that minimizes the hypervolume:

1. Non-dominated sorting of $P \cup \{s_0\}$.

2. $s \in R_I$ is discarded, where R_I is the worst ranked front.
   If $|R_I| > 1$: $\bar{s} = argmin_{s \in R_I} |\Delta_s(s, R_I)|$ with
   $\Delta_s(s, R_I) = S(R_I) - S(R_I \backslash \{s\})$

3. $R_I \longleftarrow R_I \backslash \{\bar{s}\}$

Regarding computational complexity, the hypervolume of each solution subset $R_I \backslash \{s\}$, $\forall s \in R_I$ has to be computed.

4.6.2.2 Individual-and Region-based Selection

The selection strategies of this category are individual-based as the unit of these selections are individuals ([32]). This category comprises the MOEAs PAES [95], PESA [32] and SPEA [185]. The main goal of this category is to increase the selection probability of solutions in less crowded areas and therefore the solutions spread. Hence, these selection strategies require estimations of the level of individual isolations. In PAES as well as in PESA, the objective space is divided into hyperboxes. In PESA, selection is used to choose individuals for reproduction. PESA makes use of two populations: the internal population storing the current candidate solutions, the external population or archive contains the non-dominated solutions. A 'squeeze factor' is assigned to each individual, which is the total number of solutions in the archive sharing the same hyperbox. Individuals for reproduction are only selected from the archive. Binary tournament selection is used to choose two individuals from the archive at random; the one with the lowest squeeze factor is selected.
PAES is a hillclimbing algorithm and therefore does not use a selection strategy in the common sense. Selection is performed between two solutions, the current solution and its mutant. Furthermore, the selection is used to decide if the mutant enters the archive of non-dominated solutions and if the mutant becomes the new current solution. The selection criterion is once again the squeeze factor. If the archive is not full, and if the mutant dominates the current solution and has a lower squeeze factor, the mutant enters the archive and becomes the new current solution. The selection of PAES is therefore characterized as a steady-state selection strategy.
SPEA uses a strength measure as a selection criterion. Individuals are assigned to two populations, an internal and an external one. The latter only contains the non-dominated solutions. The strength measure is determined for each individual of the external population according to the number of solutions in the internal population which is dominated. The strength measure for each indivi-

dual of the internal population is calculated by adding the strength measure of dominating individuals in the external population. The selection is performed with the focus on minimizing the strength factor. This prefers an exploration of the search process in less crowded regions of the objective space. The selection strategy in SPEA is characterized as elitist.

4.6.2.3 Direct Region-based Selection

A region-based selection strategy is introduced in PESA-II [31]. The motivation for this selection strategy is to achieve an increase of the selection probability for individuals in less crowded regions of the search space more directly. Instead of immediately selecting an individual, a hyperbox is selected by a traditional selection strategy like tournament selection. Then the preferable individual is randomly chosen from the selected hyperbox.

4.6.2.4 Crowded-comparison Operator based Selection

NSGA-II [44] uses binary tournament selection with two selection criteria: the rank of the individuals and the crowding distance. Binary tournament selection is used and individuals are selected according to the lowest rank and the highest crowding distance. Primarily, the solution with the lowest rank and therefore the fittest individual is selected. In the case that both solutions of the tournament set are in the same rank, the solution with the highest crowding distance is preferred.

4.6.3 Selection Strategies Used in VONSEA

There are several issues when designing an appropriate selection strategy for a MOEA with the aim of biochemical optimization. The first issue concerns the question of how to guide the search in the direction of the Pareto optimal solutions. The second issue is to ensure a high spread of the non-dominated solutions. The third issue is due to the specific purpose of biochemical optimization: The selection has to ensure a high diversity of the genetic material passed on the succeeding population. The high diversity of the genetic material supports the global search process. Ideally, the selection strategy has to comply with these three issues at the same time. Furthermore, another component is important for the selection process especially in the field of molecular

optimization. The role of change in the selection procedure imitates the aspect of change in a natural evolutionary process.
Two different types of selection strategies are evolved for VONSEA under three essential subjects for selection: a high diversity of genetic material within a generation, the guidance of the search process in the direction of the high quality solutions and the aspect of change. These two types of selection strategies are based on tournament selection, a combination of fitness proportionate selection and a discerning selection criterion, which is rank-based in the case of 'Aggregate Selection' and indicator-based in the case of 'ACV-based Selection' as well as 'ACV-random Selection'.

4.6.3.1 Aggregate Selection

This selection strategy is motivated by the idea of guiding the search in the direction of high-quality solutions while maintaining a high diversity of the genetic material within the succeeding generation. This strategy is tournament-based and uses a combination of front-based SUS and a rank-based discerning selection criterion. The use of tournament selection provides the subject of change in the selection process. Front-based SUS ensures the diversity of the genetic material and a potentially high solutions spread. Furthermore, it provides the opportunity for low quality solutions to find their way into the succeeding generation. Low quality solutions potentially have high quality genetic motifs, which produce high quality solutions in later generations. The rank-based discerning selection criterion ensures fitness-directional guidance. The procedure of Aggregate Selection is depicted in Fig 4.17. It starts with the tournament selection of ts individuals from the population. These individuals are ranked among themselves. From this ranked tournament set, individuals are chosen from the first front with a probability p_0 to guide the search process in direction of high quality solutions with a particular probability. With a probability $1 - p_0$, the individuals are chosen from different fronts via SUS. The number of pointers in front-based SUS is equal to the number of fronts detected in the ranking process. The segments are equal in size to the number of individuals in each front. These steps repeat until the succeeding filial generation is complete. Consequently, Aggregate Selection has two parameters, the tournament size and the probability p_0 for choosing the individuals from the first front. The fine-tuning of these parameters is subject of the experiments in order to find a good balance between high selection pressure and high diversity. The default parameters are a tournament size of 10 and a probability

of $p_0 = 50\%$.

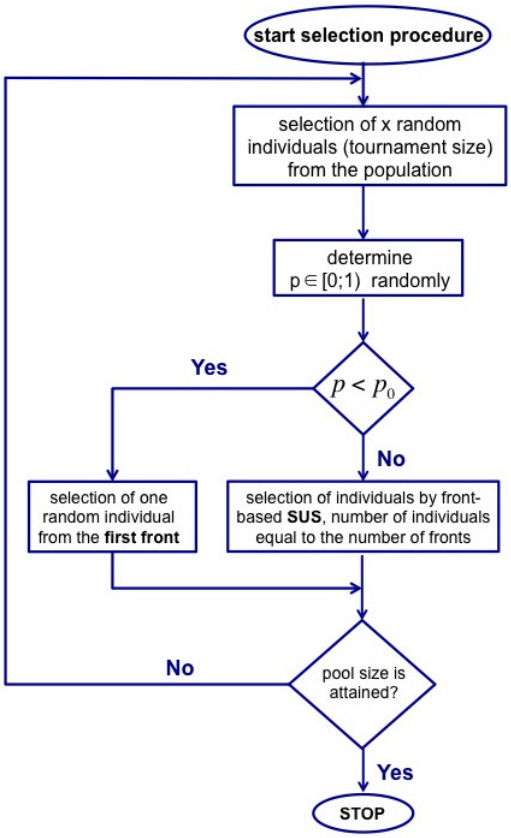

Fig. 4.17: The procedure of the aggregate selection strategy.

4.6.4 ACV-based Selection

The procedure of ACV-based selection is equal to the Aggregate Selection in the procedure, but the rank-based discerning selection is substituted by an indicator-based selection criterion. The scaled convergence metric ACV_{scaled} (eq. (5.11)) as introduced in section 5.2.2 is used as indicator to ensure indicator values that are independent of the objective scaling and ensures the selection of the fittest individual according to the objectives. The ACV-based selection proceeds as follows: The ACV_{scaled} value for each individual x_0 of the tournament set is determined with $X = \{x_0\}$ and the individual with the lowest ACV_{scaled} value is selected. The procedure is depicted in Fig. 4.18. The selection criterion differing from Aggregate Selection is highlighted. The basic idea for the ACV-based selection criterion is motivated by the following consideration of the Aggregate Selection strategy: The individuals randomly chosen by tournament selection are ranked and a random individual from the

first front is selected. The random selection of one individual from the first front does not guarantee the selection of the fittest individual with respect to all objective values, since the ranking into the first front is due to the objective values of other individuals in the tournament set. Therefore, an ACV value is determined for each individual in the tournament set: The ACV value of an individual x_0 is calculated by applying eq. (5.11) to $X = x_0$ with $n = 1$. The individual with the lowest ACV_{scaled} value is the fittest one and selected for the succeeding generation. In the case of multiple lowest ACV_{scaled} values, a random one of these is selected. The ACV-based selection strategy does not make use of a ranking method. The parameters of this strategy are once more the tournament size and the probability value p_0.

4.6.4.1 ACV-random Selection

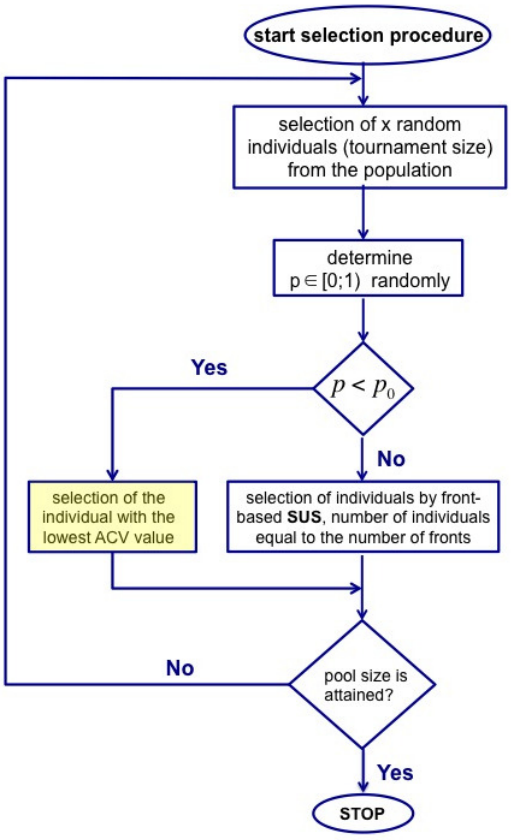

Fig. 4.18: ACV-based selection with SUS.

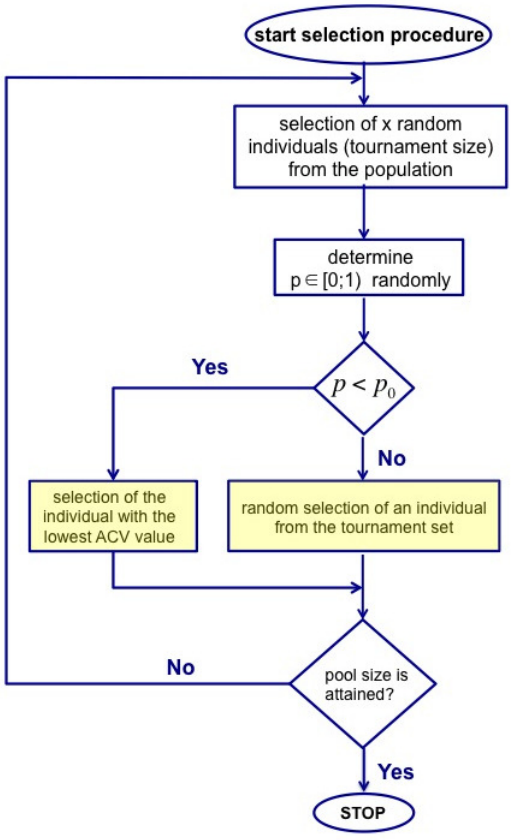

Fig. 4.19: ACV-random selection strategy.

The motivation for an alternative ACV-based selection is the empirical investigation of the influence of the fitness-proportionate selection 'front-based SUS' on the search process. The procedure of ACV-random selection is presented in

Fig. 4.19. More precisely, the comparison of the performance achieved by the VONSEA configurations with ACV-based and ACV-random selection allows the empirical conclusion on wether the fitness-proportionate selection is more promising regarding selection pressure and solution diversity than change as a selection criterion. Front-based SUS as a selection criterion is substituted by a simple random selection of an individual from the tournament set. Once again, the differing selection criteria compared to the Aggregate Selection are highlighted.

4.6.5 Computational Complexity Comparison of Selection Strategies

The selection components that are mainly responsible for the difference in computational complexity between the aggregate and the ACV-based selection are the non-dominated sorting of the tournament set and the determination of the ACV_{scaled} values for each solution in the tournament set. In the following, k is the number of objective functions and N the size of the tournament set. The computational complexity of non-dominated sorting is $O(k \cdot N^2)$ [44]. The selection of the solutions with the lowest ACV_{scaled} value starts with the determination of the maximal value for each objective: This takes $k \cdot (N - 1)$ operations for comparison. Furthermore, $k \cdot N$ divisions are performed to complete the scaling. The calculation of ACV_{scaled} for a tournament set of the size N takes k subtractions and $(k - 1)$ multiplications. The determination of the minimal ACV_{scaled} value takes $(N - 1)$ operations for comparison. In total, this procedure has a computational complexity of $k \cdot (N - 1) + k \cdot N + N \cdot (k + (k - 1)) + (N - 1) = 4kN - k - 1$ operations which is a total complexity of $O(k \cdot N)$ and therefore lower than the complexity for the non-dominated sorting.
ACV-random selection has the lowest computational complexity as the front-based SUS, which is mainly responsible for the magnitude of the computational complexity of ACV-based Selection, is replaced by the random selection of an individual.

5 Evaluation Criteria

5.1 Cost-utility Analysis of Open Source MOEA Frameworks

The aim of cost-utility analysis is the selection of a framework that allows a simple customization of the tool and a simple implementation of further components to conceive an efficient and robust MOEA for biochemical optimization. For this purpose, several criteria have been emphasized that are decisive for the selection of the framework:
The selection is focused on Java tools providing multi-objective evolutionary strategy implementations. The criterion of a Java tool is important as this allows the simplest implementation of the open source project BioJava [123] that is used for the implementation of the fitness functions. BioJava provides a set of APIs for the determination of commonly used physiochemical properties of peptide sequences composed of the canonical amino acids.
The framework has to provide the potential of multi-objective evolutionary strategies; ideally the traditional NSGA-II is preset. Moreover, the framework allows a simple extension of the implementation, which means that interfaces for the variation operators and selection method already exist. Another criterion is an intuitive program structure according to the MOEA components that supports the simple extension of the framework. The existence of a string or character encoding or at least an interface for a simple implementation of this encoding is preferable as the other MOGA components have to be adapted according to these encodings. Table 5.1 gives an overview of these criteria with the associated weightings.
Eleven open source frameworks have been qualified according to the previously mentioned criteria and are described in the following:
The framework **JAGA** (Java API for Genetic Algorithm) in its version 1.0 beta is a research tool developed and supported by the Computer Science Department of University College London (*www.jaga.org*). JAGA exhibits a

Criterion	Weights
Java framework	25 %
multi-objective evolutionary algorithms (MOEA), incl. NSGA-II	25 %
arranged program structure acc. to. MOEA comp.	20 %
character or string encoding	15 %
potential for a simple extension	15 %

Table 5.1: Overview of the weighted criteria for the selection of a suitable MOEA framework

plug-in design for simple extensibility. It provides different simple and/or elitist GAs optionally with specified initial populations. The main disadvantage of JAGA is the lack of multi-objective evolutionary algorithm implementation. Nevertheless, JAGA is qualified as it provides a protein string sequence encoding which uses 20 different characters symbolizing the 20 canonical amino acids. Each canonical amino acid is characterized in terms of the eight properties small, hydrophobic, polar, positive, negative, tiny, aliphatic and aromatic. Regarding the GA components, for each genotype JAGA contains a parameter-dependent crossover and mutation method and an elongation for amino acid patterns. The selection methods roulette wheel, tournament and two-tournament probabilistic selection are available. The user who is interested in a MOGA application has to extend this tool for this purpose, but the amino acid character encoding is a clear benefit. Moreover, JAGA has other useful functions such as BLOSUM62 (BLOcks SUbstitution Matrix) that is used for local (Smith-Waterman algorithm [146]) or global (Needleman-Wunsch algorithm [115]) sequence alignment of proteins in bioinformatics. Another useful function is the possibility to create a random initial population of protein sequences and an analysis tool. This provides a graphical and numerical analysis of each population such as the calculation of the best/worst/average fitness and standard deviation of the fitness for each generation.

The framework **jMetal** (Metaheuristic Algorithms in Java) in its version 4.5 (*jmetal.sourceforge.net*) is an extensive and complex tool especially focused on multi-objective optimization with evolutionary algorithms [50]. It includes a very large collection of metaheuristics - 20 state-of-the-art MOEAs. Therefore, jMetal provides a wide range of classical as well as recently evolved MOEAs such as NSGA-II (variants: adaptive and random NSGA-II), PESA, SPEA2, PESA2, SMS-EMOA, IBEA and MOEA-D. A graphical interface is available to support extensive experiments. Moreover, a wide set of established benchmark problems is implemented such as ZDT, WFG and DTLZ functions as well as the optimization problems of Schaffer and Scrinivas. The definite ad-

vantage of jMetal is the intuitive and clear program construction with regard to the MOEA components. Abstract classes have been defined for encoding, variation operators and selection. All in all, this framework is clearly arranged with regard to its features: metaheuristic, GA components, optimization problems, quality indicators and utilities. Within the GA components, jMetal provides abstract classes for variation operators and selection. Furthermore, different variation operators are implemented such as single-point, two-point, SBX crossover and polynomial, (non-)uniform and swap mutation. 'Ranking & crowding selection' is included as the traditional NSGA-II selection method in addition to tournament and PESA2 selection. These components are available for the genotypes integer, binary, real values and permutation. Additionally, jMetal provides several established metrics to evaluate the performance of the metaheuristics with regard to convergence and diversity. Indicators for both are implemented such as the hypervolume, inverse general distance (IGD), general distance (GD), $R2$ and measures for diversity. jMetals's disadvantage is its lack of character or string encoding and even the potential of genotype extension.

The framework **ECJ** (Java-based Evolutionary Computation Research System) in its version 21 is designed for the general purpose of evolutionary computation. ECJ is developed at George Mason University's Evolutionary Computation Laboratory (*cs.gmu.edu/∼edab/projects/ecj*) for the research purposes. Therefore, it is a stable and a most sophisticated framework. It is highly flexible with regard to the wide range of possible combinations of genotypes, methods for breeding individuals and forming a new population, fitness and selection procedures and evolutionary algorithms. The design of a specific MOEA requires an intuitive and clear program structure that lacks in the case of ECJ. A GUI system is available by a further module. Moreover, two vector representations are implemented: integer-type (byte, short, int, long) and float-type (float, double). Different variation operators are included for these vector representations such as bit-flip, uniform, polynomial and Gaussian mutation as well as one-point, two-point, uniform crossover, line recombination and SBX. Furthermore, different vector representations with corresponding variation operators are included as well as SUS and tournament selection, among others. The components vector representation, mutation and recombination allow a simple implementation of a new genotype or variation operator by overridable default methods. The ECJ implementation is mainly focused on single-objective optimization, but includes the MOEA variants NSGA-II and SPEA2.

EvA2 (Evolutionary Algorithms workbench, version 2) is a Java framework developed by the department of computer science at the Eberhard Karls Uni-

versity in Tübingen (*www.ra.cs.uni-tuebingen.de/software/EvA2*). It is not only intended for research, but is also deployed for industrial applications and is available under LGPL license. Its specificity is its easy-to-use graphical user interface which allows access to all main components. It also provides a client-server structure as well as the MOEA variants NSGA-II, PESA and SPEA2. Though the source packages are extensible, the GUI makes an extension generally rather complicated. Different variation operators are implemented optionally for the evolutionary algorithms and strategies: one-point, two-point, uniform crossover even for k parents, arithmetical and BLX-α crossover, invert-, swap-bit and Gaussian mutation. Furthermore, the selection strategies SUS, roulette and tournament selection are available. One special feature is the simple extensibility by a user-defined problem class as well as an interface for variation operators and selection by user-defined strategies. Furthermore, EvA2 provides a MATLAB interface to optimize functions in MATLAB with standard algorithm implementations in EvA2.

The framework **JCLEC** (Java Class Library for Evolutionary Computation) in its current version 4 is proposed for evolutionary computation in general. JCLEC includes the classical evolutionary algorithms NSGA-II and SPEA2. Furthermore, it provides the genotypes binary, integer and real encoding. For each genotype corresponding variation operators are implemented such as one and several loci and uniform mutation as well as one-point, two-point and uniform crossover. For real encoding the BLX-α and arithmetic crossover as well as random and non-uniform mutation are provided. SUS, tournament and roulette wheel selection are included as selection strategies. This framework has an expandable program structure as it provides abstract classes for the MOEA components mutation, recombination, selection and even genotypes. Nevertheless, its program structure is neither intuitive nor clear. With regard to program structure JCLEC is especially evolved for single-objective optimization.

MOEA framework in its current version 2.1 (*www. moeaframework.org*) is module-based and provides a wide range of MOEA variants as it includes the jMetal library in version 4.3. Therefore, classical as well as current MOEA variants such as MOEA-D, NSGA-II, IBEA, PAES, PESA-II, SMS-EMOA and SPEA2 are available. Furthermore, MOEA framework has nearly the same features as jMetal: It provides the same indicators for convergence and diversity and the benchmark problems ZDT, DTLT and WFG. A great number of different variation operators and selection strategies are also included. The main advantage of MOEA framework compared to jMetal is its capability for designing new genotypes as MOEA framework has a built-in encoding for

programs (i.e. expression tress) that is supported by some MOEA variants. The MOEAs provided by jMetal within MOEA framework only support binary, real-values and permutation encoding. However, NSGA-II is provided by MOEA framework and allows a new genotype implementation by the program elements support.

The Java framework **OPT4J** (*opt4j.sourceforge.net*) in its version 3.1 makes use of aspect-oriented programming and is a modular tool for metaheuristic optimization. It is developed under two main goals: a simple evolutionary optimization of user-defined problems and the potential of an arbitrary optimization algorithm implementation. For these purposes, it is basically modular, but provides only a limited number of MOEA implementations - NSGA-II and SPEA2. Furthermore, different benchmark problems are implemented such as ZDT, DTLZ and WFG. The genotypes integer, real, binary values and permutation are available, including a map functionality for each genotype. OPT4J offers a GUI the algorithm configuration. The modular program structure and the use of the GUI makes an extension of this framework regarding a new encoding more complicated. A special feature of OPT4J is the graphical visualization of the optimization process in form of a convergence and a Pareto plot.

Evolving Objects is an evolutionary computation framework written in C++ (*eo.dev.sourceforge.net*). Its version is 1.3.1. It is developed by a project team around Maarten Keijzer and Marc Schoenauer. The special feature of this framework is that it provides no explicit NSGA-II implementation, but NSGA-II is able to be designed by building the required structure from the following blocks: initialization, variation, selection, stopping criteria, replacement, evaluation. A clear advantage is the potential of writing a user-defined encoding beneath the state-of-the-art encodings. Furthermore, several selection strategies like rank-based, stochastic tournament, roulette and elitist are available. Moreover, variation operators as well as statistics for the purpose of evaluation and a graphical display are provided.

ParadisEO is a metaheuristic framework written in C++ for MOO based on evolving objects in its version 2.0 (*paradiseo.gforge.inria.fr*). ParadisEO is developed by the DOLPHIN project team of Inria Lille. It provides the metaheuristics MOGA, NSGA(-II), SPEA2 and IBEA. Beneath the usual encodings, a user-defined genotype is creatable. Moreover, the well-known benchmark problems ZDT, DTLZ and WFG are provided as well as statistic tools to measure diversity and convergence like the hypervolume and the Euclidean distance. It lacks an intuitive and clear program structure or the state-of-the-art variation operators. But these are available via the implementation of external

tools termed 'problem repository', which include different operators and the mentioned benchmark problems. Consequently, these variation operators are genotype-dependent.
The framework **Heuristic Lab** (A paradigm - Independent and Extensible Environment for Heuristic Optimization, version 3.3.10) is written in C♯ and won the second place at the Microsoft Innovation Award 2009. It is developed by a project team around Stefan Wagner (*dev.heuristiclab.com*). A NSGA-II implementation is available. It provides a graphical algorithm design to modify an algorithm for interactive algorithm development, analysis, application of the heuristic method and possibility of solving user-defined problems. Its implementation is strongly focused on the GUI and therefore makes a customization very complicated.
The framework **Open BEAGLE** (a generic evolutionary computation framework, version 3.0.3) is written in C++ and is developed by the 'Laboratoire de vision et Systѐme numériques' (*code.google.com/p/beagle*). It provides the following state-of-the-art genotypes: bit string, integer- and real-value, permutation. An implementation of NSGA-II is available for float values as well as different standard variation operators. BEAGLE lacks of the arranged program structure and the implementation of further genotypes is complicated. It is interesting that the same developer team provides a similar framework written in Python, termed DEAP.

The features of the frameworks' criteria are assessed with the help of a point system from 0 (missing) to 5 (completely available). Points between 1 and 4 denote a partial availability in an ascending level. Tables 5.2 and 5.3 evaluate the alternative frameworks and constitute a ranking of these. The tables reveal that none of the open source Java frameworks attain all required aspects. The frameworks jMetal and 'MOEA framework' achieve the highest scoring. As jMetal is a part of 'MOEA framework', the experiments presented in this thesis are performed with an implementation of VONSEA within jMetal. Nevertheless, some programming effort is necessary regarding this implementation. Furthermore, the protein string sequence encoding of JAGA serves as a model for the targeted peptide encoding.

Criteria	w	JAGA p	JAGA score	jMetal p	jMetal score	ECJ p	ECJ score	EvA2 p	EvA2 score	JCLEC p	JCLEC score	MOEA fr. p	MOEA fr. score
Java framework	0.25	5	1.25	5	1.25	5	1.25	5	1.25	5	1.25	5	1.25
MOEA, incl. NSGA-II	0.25	0	0	5	1.25	5	1.25	5	1.25	5	1.25	5	1.25
arr. program structure acc. to. MOEA comp.	0.2	4	0.8	5	1	1	0.2	1	0.2	2	0.4	3	0.6
character or string encoding	0.15	5	0.75	1	0.15	2	0.3	1	0.15	2	0.3	4	0.6
potential for a simple extension	0.15	1	0.15	4	0.6	2	0.3	1	0.15	3	0.45	4	0.6
total score	1		2.95		4.25		3.3		3.15		3.5		4.3

Table 5.2: Scoring table I of the alternative frameworks providing (multi-objective) evolutionary strategy implementations. (w = weight, p = points)

Criteria	w	OPT4J p	OPT4J score	EO p	EO score	ParadisEO p	ParadisEO score	Heuristic LAB p	Heuristic LAB score	BEAGLE p	BEAGLE score
Java framework	0.25	5	1.25	0	0	0	0	0	0	0	0
MOEA, incl. NSGA-II	0.25	5	1.25	3	0.75	5	1.25	5	1.25	5	1.25
arr. program structure acc. to. MOEA comp.	0.2	1	0.2	1	0.2	3	0.6	1	0.2	1	0.2
character or string encoding	0.15	1	0.15	4	0.6	4	0.6	0	0	0	0
potential for a simple extension	0.15	1	0.15	3	0.45	2	0.3	1	0.15	1	0.15
total score	1		3		2.4		2.75		1.6		1.6

Table 5.3: Scoring table II of the alternative frameworks providing (multi-objective) evolutionary strategy implementations. (w = weight, p = points)

5.2 Metrics for Convergence and Diversity

The evaluation of the performance of MOEAs has to fulfill specific criteria. The evolution of performance measures is a complicated issue as the results of a MOEA are provided as a set of vectors and not single values, which are statistically easy to analyze. Theoretically, any two vectors have to be compared component-wise under the aspect of domination. The aim is to propose a measure which compares non-dominated solution sets by mapping the quality of such a set into a single value. The criteria for such an appropriate measure has been suggested by Zitzler [181]:

1. The distance of the approximate non-dominated solution set to the Pareto optimal front has to be minimized.
2. The distribution of the non-dominated solution set over the solution space is desirably uniform.
3. The extension of the approximated non-dominated solution front is desirably maximal. This means that for each objective function a wide range of function values are present.

Therefore, two types of criteria are established to representatively evaluate the performance of a MOEA: diversity and convergence metrics. For both criteria, a wide range of metrics have been proposed in the past. Diversity metrics have been introduced, which assess the distribution of non-dominating solutions by calculating the average distance of the solutions. Convergence metrics have been presented, which compare the non-dominating solution sets with the true Pareto front (PF_{true}), respectively with an approximation set of Pareto optimal solutions (PF_{approx}). The measure of a population's convergence is also an established method to terminate or re-start a MOEA.

5.2.1 Review of Convergence Metrics

In the following, the established measures that reflect the convergence behavior of a population to PF_{true} or PF_{approx} are presented.

The S-metric or Hypervolume Metric. The hypervolume [184] is also known as S-metric [180] and is equivalent to the Lebesgue measure [102]. It measures the space spanned by the non-dominated solutions to a pre-defined

anti-optimal point r, where doubly overlapping regions are only counted once. Therefore, the more the solutions uniformly approximate PF_{true}, the more the metric value increases. Hence, the hypervolume determines the convergence of a Pareto optimal solution set as well as the diversity of the non-dominated solutions in the search space.

Definition 8 *Let $\{v^{(1)}, v^{(2)}, ..., v^{(n)}\} \subset \mathbb{R}^d$ define a finite set of elements and $r \in \mathbb{R}^d$ the anti-optimal point with $v^{(i)} \prec r$ for all $i = 1, ..., n \in \mathbb{N}$. The quantity*

$$H(v^{(1)}, ..., v^{(n)}; r) = Leb(\cup_{i=1}^{n}[v^{(i)}, r]) \tag{5.1}$$

is termed S-metric or hypervolume metric in $\mathbb{R}^d$.

For $d = 2$ with the elements $v^{(1)}, v^{(2)}, ..., v^{(n)}$ ascendingly ordered, i.e. $v^{(1)} < v^{(2)} < ... < v^{(n)}$ equation (5.1) arises to

$$H(v^{(1)}, ..., v^{(n)}; r) = (r_1 - v_1^{(1)})(r_2 - v_2^{(1)}) + \sum_{i=2}^{n}(r_1 - v_1^{(i)})(v_2^{(i-1)} - v_2^{(i)}). \tag{5.2}$$

The hypervolume is one of the most established metrics because of its favorable mathematical properties [186]. One disadvantage of this operator is the choice of the anti-optimal point as it influences the results and is the subject of ongoing research. Other disadvantages are its sensitivity to the relative scaling of the objectives, the presence or absence of extreme points of a front and the high computational complexity caused by the necessary point ordering [54]. Furthermore, this metric prefers convex regions to non-convex ones [184]. The hypervolume indicates the closeness to PF_{true} and diversity [185]. A great amount of research has been done to find an implementation of the hypervolume with reduced computational complexity. An overview of established algorithms for calculating the hypervolume with the worst-case computational complexities is given in Table 5.4. Short descriptions and pseudo-codes of some of these algorithms are presented in the work of Bradstreet et al. [20].

D-metric. Another convergence metric is the D-metric introduced by Zitzler [181]. Starting point are two sets of Pareto optimal solutions A and B. This metric calculates the size of the space dominated by A and not dominated by B.

$$D(A, B) = H(A + B; r) - H(B; r),$$

where $H(A; r)$ denotes the hypervolume with the anti-optimal point r. A reference set of PF_{true} is needed to use this metric as convergence measure for a set of non-dominated solutions.

Algorithm	**Comput. complexity**	**Ref.**
Inclusion-Exclusion	$O(k \cdot 2^n)$	[175]
LebMeasure	$O(n^k)$	[172]
Hypervolume by Slicing Objectives (HSO)	$O(n^{k-1})$	[172]
Optimal 3D Hypervolume	$O(n \log(n))$	[16]
Fonseca Paquete López-Ibánez (FPL)	$O(n^{k-2} \cdot \log(n))$	[67]
Hypervolume Overmars and Yap (HOY)	$O(n \cdot \log n + n^{d/2})$	[29]
Brinkmann	$O(n^{(k+2)/3})$	[22]
Yildiz and Suri	$O(n^{(k-1)/2} log(n))$	[177]
Walking Fish Group (WFG)	$O(2^n)$	[171]

Table 5.4: Algorithms for calculating the hypervolume with reduced computational complexity. The number of non-dominated solutions is n, and k is the number of objectives.

Set Coverage Metric (C-metric). Zitzler also proposed the C-metric [184], which is an appropriate measure to compare the dominance of two Pareto optimal sets PF_1 and PF_2. The C-metric maps the ordered pair $(PF1, PF2)$ into the interval $[0; 1]$:

$$C(PF_1, PF_2) := \frac{\mid \{b \in PF_2 \mid \exists a \in PF_1 : a \preceq b\} \mid}{\mid PF_2 \mid} \tag{5.3}$$

Therefore, the value $C(PF_1, PF_2) = 0$ means that no solution of PF_2 is weakly dominated by at least one solution of PF_1, whereas $C(PF_1, PF_2) = 1$ implicate that all points of PF_2 are weakly dominated by PF_1. This metric is usually not symmetric, therefore $C(PF_1, PF_2)$ is not a metric in a mathematical sense and consequently $C(PF_1, PF_2)$ and $C(PF_2, PF_1)$ have to be determined.

Error Ratio. The Error Ratio (ER) [159] is introduced by Veldhuizen and is a percentage measure for the number of solutions in a set that lies on PF_{true}. PF_{approx} is used as a reference set of Pareto optimal solutions in this metric.

$$ER(PF_{approx}) = \frac{1}{|PF_{approx}|} \sum_{i=1}^{|PF_{approx}|} e_i \text{ whereas} \tag{5.4}$$

$$e_i = \begin{cases} 0 & \text{if the solution vector } i \text{ is in } PF_{approx} \\ 1 & \text{if the solution vector } i \text{ is not in } PF_{approx} \end{cases} \tag{5.5}$$

A measure of $ER \approx 1$ means that PF_{approx} comprises only a low number of solutions in PF_{true}, whereas a lower measure value indicates that many solutions are in PF_{true}. ER is exceptionally sensitive to the reference set PF_{true}:

If a Pareto optimal solution is not in PF_{true}, it is treated as a non-optimal solution by ER. Furthermore, ER does not take the closeness of PF_{approx} to PF_{true} into account.

General Distance. Generational Distance (GD) is also proposed by Veldhuizen [160]. This metric is a measure of the average distance between solutions of PF_{approx} and PF_{true} and is defined as:

$$GD(PF_{approx}) = \frac{\left(\sum_{i=1}^{n} d_i^p\right)^2}{n},$$

where n is the number of solutions in PF_{approx}, usually $p = 2$ and d_i is the Euclidean distance between each solution in PF_{approx} to its nearest member in PF_{true}. A value of $GD(PF_{approx}) = 0$ denotes that $PF_{approx} = PF_{true}$. However, the GD provides no information about homogeneity, spread or dominance of PF_{approx} compared to PF_{true}.

Convergence Metric of Deb. Deb [41] proposes a convergence metric that evaluates the distance of PF_{approx} to a reference set of PF_{true}, further denoted as PF^*. $PF^* = \{a_1, a_2, ..., a_n\}$ is either a solution set of the Pareto optimal front or an approximate Pareto optimal set obtained from previous MOEA runs. In each generation, the following steps have to be performed for the determination of this metric:

- Generate the non-dominated solution set $PF_{approx} = \{p_1, p_2, ..., p_n\}$.

- The smallest normalized Euclidean distance d_i for each solution of PF_{approx} to PF^* is calculated via:

$$d_i = \min_{j=1,...,n} \sqrt{\sum_{k=1}^{M} \left(\frac{f_k(a_i) - f_k(p_j)}{f_k^{max} - f_k^{min}}\right)^2},$$

  where M denotes the number of objective functions, f_k^{max} is the maximal and f_k^{min} is the minimal function value of the k-th objective function of PF^*.

- The convergence metric value is determined as the average normalized distance for all solutions in PF_{approx}

$$C(PF_{approx}) = \sum_{i=1}^{|PF_{approx}|} \frac{d_i}{|PF_{approx}|} \tag{5.6}$$

The lower the values for this metric the better the convergence.

Averaged Hausdorff Distance. This recently proposed performance measure for convergence is introduced by Schütze [140]. The Averaged Hausdorff Distance (Δ_p) is a combination of the slightly modified GD [163] and the Inverted General Distance (IGD) [30]. Δ_p is defined by:

$$\Delta_p(X_Y) = \max(GD(X,Y), IGD(X,Y)) \tag{5.7}$$

$$= \max\left(\frac{1}{m}(\sum_{i=0}^{m} dist(x_i,Y)^p)^{1/p}, \frac{1}{n}(\sum_{i=0}^{n} dist(y_i,X)^p)^{1/p}\right) \tag{5.8}$$

with the finite non-empty sets $X = \{x_1, x_2, ..., x_m\}$ and $Y = \{y_1, y_2, ..., y_n\}$, where X is regarded as a set of approximate Pareto optimal solutions and $Y = PF_{true}$.

R2 Indicator. Trautmann [156] recently proposed the $R2$ indicator that evaluates the quality of PF_{approx} regarding the convergence to PF_{true}, the solutions spread and the representation of the Pareto front shape. The R2 indicator is defined by:

$$R2(S,W,r) = \frac{1}{N}\sum_{w\in W} \min_{s\in S} \max_{j} w_j \cdot (s_j - r_j), \tag{5.9}$$

where $W = \{w^1, ..., w^N\} \subset \mathbb{R}^k$ is a set of N weight vectors, $S \subset \mathbb{R}^k$ a set of solutions and $r \in \mathbb{R}^k$ is an ideal point that usually is chosen as an optimal objective vector better than all feasible solutions.
This indicator is popular for its computational complexity $O(Nk \cdot |S|)$ indicating that the complexity is linear with the number of weights, the problem dimension and the number of the solution sets. The number of weight vectors is an open issue, especially for $k > 3$. The volume of the space increases exponentially with k and potentially also the number of weight vectors. This makes the calculation of R2 from a specific number of k on as expensive as the hypervolume. [168]
Empirical results have shown that the R2 indicator and the hypervolume are

correlated by Pearson's correlation coefficient with a statistically significant value of 0.76 [23].

5.2.2 A Statistical Indicator as a Convergence Metric: The Average Cuboid Volume

For the purpose of evaluating the VONSEA populations in terms of convergence, a statistical indicator is introduced. There are three major aspects for the use of the Average Cuboid Volume (ACV) as a convergence indicator: Firstly, the disadvantage of the established metrics D-metric, ER, GD, Δ_p and of the convergence metric of Deb is the requirement of the knowledge of PF_{true} or at least a reference set of Pareto optimal solutions that are usually unknown in real-world MOPs and also for the presented biochemical optimization problems in this thesis. Secondly, this indicator measures the quality of the solutions set relative to the set size. As a consequence, this indicator allows an entire ranking between populations of different sizes in a statistically reasonable way. Thirdly, another reason for using a statistical measure as a convergence indicator is the aim of evaluating the convergence progress of the entire population. Intuitive hints for this purpose are the distances of the different fronts to the true Pareto front and the changes of the different front sizes through the generations. This requires an adequate metric mapping of the information into one value is missing. The proposed statistical measure provides the information of the population progress through the generations. Therefore, the proposed indicator is intended to measure the quality of the entire population, not only of a non-dominated solution set.
The indicator calculates the average cuboid volume of the cuboids spanned by the solution points of an entire population to a pre-defined ideal point r. ACV is defined by eq. (5.10). This ideal point is chosen as a theoretical optimal point of eq. (1.1). In many MOPs, it is easier to find an optimal point than an anti-optimal one - especially in the case of the multi-dimensional biochemical minimization problems. In contrast, the ideal point is the objective vector that is simply better in each objective than all feasible solutions.

$$ACV(X) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right), \tag{5.10}$$

where n is the population size, k is the number of objectives, x_i are the solutions on the population X and x_{ij} is the $j-th$ component of a solution x_i. It is $x_{ij} - r_j \geq 0$ as r_j is assumed to be the theoretical optimal point of a

minimization problem. The lower the indicator values, the better the global quality of the solution set X, since the ideal point is chosen as a theoretical optimal point.

ACV is evolved according to the model of the hypervolume, but differs from the hypervolume in several aspects: ACV and the hypervolume measure the space covered by the solutions to a pre-defined point. This point is chosen as an optimal point in the case of ACV rather than an anti-optimal one in the case of the hypervolume. The covered spaces of the solutions to this point are averaged in the case of ACV, whereas multiple covered space is counted only once in the case of the hypervolume. As a consequence, adding a dominating solution to a set of non-dominated solutions, where the objective values of the dominating solution are not extremal compared to those of the non-dominated solutions, does not modify the hypervolume value. Moreover, as the hypervolume does not take the number of solutions into account, it is not a statistical measure and does not allow a direct comparison of differently sized solutions sets. In terms of convergence, the overall quality of a population is better the lower the ACV value, whereas the quality of a non-dominated set is better the higher the hypervolume value. Furthermore, the computational complexity of ACV is lower than that of the hypervolume even if the number of objectives increases, as no point ordering is required. For each solution, k subtractions and $(k-1)$ multiplications have to be performed. Therefore, the computational complexity of ACV for a solution set of n individuals and k objectives amounts to $O(n \cdot k)$.

A normalized version of the ACV indicator is proposed to ensure that all objective function values have the same influence on the indicator values and to avoid potential problems with the scaling of the objective space. A different scaling of the objectives results in a different influence of the objective values on the indicator values and has been stated as a point of criticism with regard to the hypervolume indicator [54]. Therefore, a further ACV-based indicator is proposed, where every objective function value is divided by the maximum norm. Therefore, ACV_{scaled} maps all objective values in the same range of $[0; 1]$:

$$ACV_{scaled} = \frac{1}{n} \sum_{i=1}^{n} \left(\prod_{j=1}^{k} \frac{(x_{ij} - r_j)}{\bar{x}_j} \right), \text{ with } \bar{x}_j = max_i\{x_{ij} - r_j\}, \ \forall j = 1, ..., k \tag{5.11}$$

A relative ACV indicator is proposed to evaluate the average cuboid volume of the first front solutions relative to the average cuboid volume of the entire

population:

$$ACV_{rel} = \frac{\frac{1}{f}\sum_{i=1}^{f}(\prod_{j=1}^{k}(x_{ij} - r_j))}{\frac{1}{n}\sum_{i=1}^{n}(\prod_{j=1}^{k}(x_{ij} - r_j))}, \tag{5.12}$$

where f is the number of solutions in the first front. A very small value of ACV ($ACV \approx 0$) indicates that ACV of the first front is much smaller than ACV of the entire population and a particular number of high quality peptides have been identified. In the case of $ACV_{rel} \approx 1$, the ACV value of the first front is relatively high compared to the ACV value of the entire population. A further interpretation of the relative ACV values has to take account of the absolute ACV of the entire population.

5.2.2.1 Discussion of $ACV(X)$

The suitability of a metric for evaluation depends on the intention of the investigation object and analysis preferences. ACV is intended to evaluate the global convergence behavior of an entire population with the ultimate aim of comparing solution sets of different sizes in a statistically reasonable way according to the proximity to PF_{true}.
The first important aspect in favor of the use of ACV is that convergence quality does not change in the case of multiple copies of one solution. ACV does not fulfill this averaging strategy that can be manifested through the following example: Let $x \in \mathbb{R}^k$ be a solution of equation (1.1). Furthermore, $Y = \{x, x, ..., x\}$ is a bag containing n copies of the solution x, then

$$ACV(Y) = \frac{1}{n}\sum_{i=0}^{n}\left(\prod_{j=1}^{k}(x_j - r_j)\right) = \frac{1}{n}\cdot n\prod_{j=1}^{k}(x_j - r_j) = \prod_{j=1}^{k}(x_j - r_j) = ACV(X)$$

The second aspect is due the following observation: An intuitive indicator reflecting the quality of approximation sets of different Pareto front refinements requires 'better' indicator values for the finest approximation set. The following example demonstrates this effect for ACV:

Example 5 *The Pareto front is given by the bounded convex function* $f(x) = 1/x^2$ *between the points* $y_1 = (0.1, 100)$ *and* $y_2 = (1.1, 0.826)$ *meaning*

$$PF_{true} = \{(x, y) | y = 1/x^2 \text{ with } x \in [0.1, 1.1]\}. \tag{5.13}$$

We consider the following three approximation sets of increasing refinement of the Pareto front

$$Y_1 = \{(0.1 + 0.2 \cdot i, 1/(0.1 + 0.2 \cdot i)^2) | i \in \{0, 1, ..., 5\}\}, \tag{5.14}$$

$$Y_2 = \{(0.1 + 0.1 \cdot i, 1/(0.1 + 0.1 \cdot i)^2) | i \in \{0, 1, ..., 10\}\}, \tag{5.15}$$

$$Y_3 = \{(0.1 + 0.01 \cdot i, 1/(0.1 + 0.01 \cdot i)^2) | i \in \{0, 1, ..., 100\}\}. \tag{5.16}$$

Table 5.5 depicts the indicator values of ACV for the three approximation sets with the ideal point $(0, 0)$.

These results are reproducible for a concave function if the ideal point is chosen as the theoretical maximal limit of the objective functions.
The third aspect of this indicator is the averaging effect. It is obvious that a dominating solution x yields better indicator values than the dominated solution y, because $ACV(\{x\}) = \prod_{i=1}^{k}(x_j - r_j) < \prod_{i=1}^{k}(y_j - r_j) = ACV(\{y\})$. This observation allows the interpretation that if one dominated solution x_1 in the solution set $X = \{x_1, x_2, ..., x_n\}$ is replaced by a dominating one $\bar{x}_1$, then $ACV(\{x_1, x_2, ..., x_n\}) > ACV(\{\bar{x}_1, x_2, ..., x_n\})$. The averaging effect is illustrated by the following example [140]:

Example 6 *The true discrete Pareto front is described by* $P = \{p_i | p_i = (0.1 \cdot (i-1); 1-(i-1) \cdot 0.1)$ with $i = 1, ..., 11\}$. *Two solution sets are given by* $X_1 = \{x_{1,1}, p_2, ..., p_{11}\}$ *and* $X_2 = \{x_{2,1}, x_{2,2}, ..., x_{2,11}\}$ *with the elements* $x_{1,1} = (\epsilon, 10)$ *and* $x_{2,i} = p_i + (\frac{\epsilon}{2}, 5)$ *with* $i = 1, ..., 11$. *For the outlier* $x_{1,1}$ *the values* $\epsilon = 0.001$ *is used for numerical evaluations.* X_1 *is a better approximation of the true Pareto front than* X_2 *as all solutions exceeding the outlier* $x_{1,1}$ *are positioned on the Pareto front. All points of* X_2 *are shifted by* $(\frac{\epsilon}{2}, 5)$ *from the Pareto front, but the difference of each element to* PF_{true} *is less than the outlier* $x_{1,1}$. *As we are interested in an averaging effect, the indicator values of* X_1 *have to be better than the one of* X_2. *This is true for* $ACV(X)$ *as* $ACV(X_1) = 0.15$ *and* $ACV(X_2) = 2.65$ *with the ideal point* $(0, 0)$.

Apart from these preferences, ACV fulfills the important complement property of location parameters [166]. This complement property is formulated as an axiom:

X	Y_1	Y_2	Y_3
ACV(X)	3.13	2,75	2.43

Table 5.5: Indicator values of ACV for approximation sets $Y_1 - Y_3$ with ideal point $(0, 0)$.

Axiom 1 *Given are n values $x_1, x_2, ..., x_n$ with the location parameter M_n. In the case that a further value x_{n+1} enters the set, the following statements hold for the new location parameter $M(\{x_1, x_2, ..., x_{n+1}\}) = M_{n+1}$:*
if $x_{n+1} \geq M_n$, then $M_{n+1} \geq M_n$;
if $x_{n+1} \leq M_n$, then $M_{n+1} \leq M_n$

The complement property is important for the robustness of a measure and this property is further proven for the ACV indicator regarding the comparison of two solution sets:

Proposition 1 *Given are two solutions sets $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_{m+l}\}$ with $m, n, l \in \mathbb{N}$ and it holds:*
(i) $\forall i \in \{1, ..., n\}$, $\forall j \in \{1, ..., m\}$ $y_j \preceq x_i$ and
(ii) $\forall i \in \{1, ..., n\}$, $\forall j \in \{m+1, ..., m+l\}$ $y_j \prec x_i$
Then $ACV(Y) < ACV(X)$.

Proof: It has to be shown that

$$\begin{aligned} ACV(X) > ACV(Y) &\Leftrightarrow \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right) > \frac{1}{m+l}\sum_{i=1}^{m+l}\left(\prod_{j=1}^{k}(y_{ij} - r_j)\right) \\ &\Leftrightarrow (m+l)\sum_{i=1}^{n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right) > n\sum_{i=1}^{m+l}\left(\prod_{j=1}^{k}(y_{ij} - r_j)\right) \end{aligned} \tag{5.17}$$

It holds

$$(m+l)\sum_{i=1}^{n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right) \geq (m+l)\cdot n \cdot \min_{i=1,...,n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right).$$

According to the conditions (i) and (ii), it holds

$$(m+l)\min_{i=1,...,n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right) \geq (m+l)\max_{i=1,...,m+l}\left(\prod_{j=1}^{k}(y_{ij} - r_j)\right) > \sum_{i=1}^{m+l}\left(\prod_{j=1}^{k}(y_{ij} - r_j)\right)$$

From this inequalities eq. (5.17) is proven:

$$(m+l)\sum_{i=1}^{n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right) \geq (m+l)\cdot n\cdot \min_{i=1,...,n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right) > n\cdot\sum_{i=1}^{m+l}\left(\prod_{j=1}^{k}(y_{ij} - r_j)\right)$$

q.e.d.

The use of $ACV(X)$ as a convergence and as a diversity metric is not in the focus since $ACV(X)$ is not a reliable indicator for diversity. A solution set with clustered solutions does not always achieve worse indicator values than a less clustered solution set. This is demonstrated by the following example:

Example 7 *Once more* PF_{true} *is described by equation (5.13) and the solution set* $Y_4 = \{(0.29, 11.89), (0.3, 11.11), (0.31, 10.4), (0.32, 9.77), (0.33, 9.18), (0.34, 8.65)\}$ *contains clustered solutions on the true Pareto front, then* $ACV(Y_4) = 3.18 \approx ACV(Y_1)$ *Though the solutions of* Y_4 *are much more clustered than those of* Y_1*,* Y_4 *receive nearly the same indicator values as* Y_1*.*

Moreover, this effect has also been empirically investigated. Therefore, the Pearson's correlation coefficient [99] has been determined between the ACV and diversity values (calculated by eq. (5.24)) arising from the evaluation of 400 populations from VONSEA test runs. A coefficient of 0.24 is achieved indicating only a very weak correlation between the diversity and the ACV values.

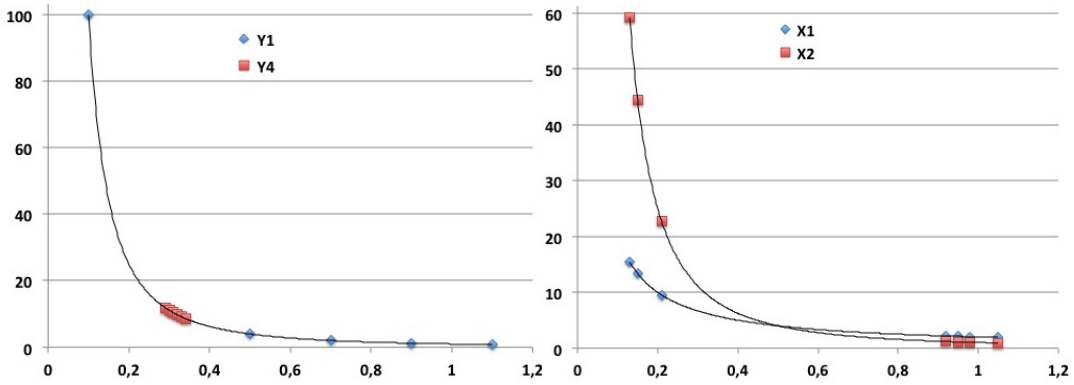

Fig. 5.1: Visualization of Example 3 (left figure) and Example 4 (right figure).

As a consequence, the use of the ACV indicator as a selection criterion results in very clustered solutions on one part of the Pareto front and makes a further diversity preserving method necessary. This effect is demonstrated by the following simple example.

Example 8 *Two Pareto fronts are given by the bounded convex functions* $f(x) = 2/x$ *and* $g(x) = 1/x^2$ *between the x-coordinates* 0.1 *and* 1.1*, meaning*

$$PF1_{true} = \{(x, y) | y = 2/x \text{ with } x \in [0.1, 1.1]\} \tag{5.18}$$

$$PF2_{true} = \{(x, y) | y = 1/x^2 \text{ with } x \in [0.1, 1.1]\} \tag{5.19}$$

We consider an approximation set for each Pareto front: X_1 is an approximation set for $PF1_{true}$ and X_2 is one for $PF2_{true}$. The solutions are each positioned at the boundaries of the Pareto fronts.

$$X_1 = \{x_1(0.13, 15.38), x_2(0.15, 13.33), x_3(0.21, 9.52), x_4(0.92, 2.17), x_5(0.95, 2.11), x_6(0.98, 2, 04), x_7(1.05, 1.91)\},$$

$$X_2 = \{x_1(0.13, 59.17), x_2(0.15, 44.44), x_3(0.21, 22.68), x_4(0.92, 1.18), x_5(0.95, 1.11), x_6(0.98, 1.04), x_7(1.05, 0.91)\},$$

Table 5.6 depicts the indicator values of each solution in the approximation sets determined with the ideal point $(0, 0)$.

In the case of the approximation set X_1, all solutions have the same probability to be chosen for reproduction. In contrast, the solutions at the right boundary of X_2 are preferred by the selection strategy based on the ACV indicator. In conclusion, if the solutions on the Pareto front do not exhibit the same ACV values, the search process is guided in the direction of the lowest ACV values and therefore results in clustered solutions on one part of the Pareto front.
The ACV_{rel} indicator is based on the ACV to measure the quality of the non-dominated solutions relative to the convergence of the entire population. Though the ACV_{rel} indicator is quite different in its significance from the hypervolume indicator, since the quality of the non-dominated solutions in the latter case is not related to the quality of the entire population. For a deeper insight into the relation of ACV_{rel} to the standard quality measure hypervolume that is commonly used to evaluate the quality of the non-dominated solutions, ACV_{rel} is statistically compared to the hypervolume by Pearson's correlation coefficient. For this purpose, correlation coefficients are determined out of ACV_{rel} and hypervolume values from 200 populations of VONSEA test runs to assess statistical significance and a value of 0.6 is achieved indicating a moderate correlation.

ACV(X)	x_1	x_2	x_3	x_4	x_5	x_6	x_7
X_1	2	2	2	2	2	2	2
X_2	7.7	6.66	4.76	1.1	1.05	1.02	0.95

Table 5.6: ACV value for each solution of the approximation sets with the ideal point $(0, 0)$.

5.2.3 Review of Diversity Metrics

Spacing Metric of Deb. The Spacing metric of Deb [44] measures the spread of the obtained N non-dominated solutions and characterizes the homogeneity and the evenness of the solutions' distribution over the solution space:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \mid d_i - \bar{d} \mid}{d_f + d_l + (N-1)\bar{d}}, \tag{5.20}$$

where d_i denotes the Euclidean distance between two consecutive solutions of the non-dominated set and $\bar{d}$ is the average distance of all these distances. d_f, d_l are the Euclidean distances between the extreme (feasible) solutions and the boundary solutions of the non-dominated solutions. In general, the diversity of the solutions is better the smaller Δ. Therefore, the widest and most uniform solutions spread is reached in the case that $d_i \approx \bar{d}\ \forall i$ and $d_f = d_l = 0$. The main disadvantage of this metric is that it is only suitable for two-dimensional objective spaces as consecutive solutions need a more sophisticated definition in higher dimensions. This metric provides no information about convergence, hence a second metric is needed to evaluate the performance of a MOEA.

Spacing Metric of Schott. Another spacing metric has been introduced by Schott [138] which determines how evenly the points in a finite approximate Pareto optimal set PF_{approx} are distributed over the solution space:

$$\Delta_s(PF_{approx}) = \sqrt{\frac{1}{\mid PF_{approx} \mid -1} \sum_{i=1}^{\mid PF_{approx} \mid} (\bar{d} - d_i)^2}, \text{ where} \tag{5.21}$$

$$d_i = \min_j \{ \sum_{m=1}^{M} \mid f_m(a_i) - f_m(a_j) \mid : a_i, a_j \in PF_{approx}; i, j = 1, ..., \mid PF_{approx} \mid \}, \tag{5.22}$$

where $\bar{d}$ denotes the average distance of all d_i and M is the number of objective functions. A benefit of this metric is its low computational cost and the fact that it is suitable for all dimensions. A value of $\Delta_s(PF_{approx}) \approx 0$ indicates that all non-dominated solutions are equidistant in the solution space.

Region-based Diversity Metric of Deb. Deb proposes a region-based diversity metric providing a fast calculation of the solutions spread. The main idea of this metric is to project the non-dominated solutions of a generation

on a suitable hyper-plane and therefore to reduce the solution dimension. This hyper-plane is divided into (k-1) dimensional boxes, where k is the number of objectives. Diversity is calculated by determining whether each grid cell contains one of the non-dominated solutions or not. Optimal diversity is reached if all grid cells contain at least one non-dominated solution. The following steps present the procedure to determine diversity [41]. P^* denotes a set of Pareto optimal solutions, $P(t)$ the population of generation t and $F(t)$ the non-dominated solution set of the current generation:

1. Determination of $F(t)$ from $P(t)$ with $F(t)$ is non-dominated to P^*.

2. For each grid denoted by $(i, j, ...)$ calculate the arrays:

$$H(i,j,...) = \begin{cases} 1 & \text{if the grid has a representative point in } P^* \\ 0 & \text{otherwise} \end{cases}$$

$$h(i,j,...) = \begin{cases} 1 & \text{if } H(i,j,...) = 1 \text{ and the grid has a representative point} \\ & \text{in } F(t) \\ 0 & \text{otherwise} \end{cases}$$

3. Assignment of a value $m(h(i, j, ...))$ to each grid depending on its and its neighbor's $h(i, j, ...)$. Similarly, calculate $m(H(i, j, ...))$ using $H(i, j, ...)$ as reference point.

4. Calculation of the diversity metric by averaging the individual $m(h(i, j, ...))$ values with respect to $m(H(i, j, ...))$:

$$D(P(t)) = \frac{\sum\limits_{\substack{i,j,... \\ H(i,j,...)\neq 0}} m(h(i,j,...))}{\sum\limits_{\substack{i,j,... \\ H(i,j,...)\neq 0}} m(H(i,j,...))} \tag{5.23}$$

5.2.4 A New Diversity Metric: The Average Spacing Metric

The Average Spacing Metric (ASM) is motivated by the diversity metric of Deb. The main disadvantage of Deb's metric is its limited use regarding the dimension of the search space. The determination of Euclidean distances between two consecutive solutions is only possible for a two-dimensional search space. Two aspects are important for the motivation of ASM: Firstly, it has to be suitable for higher dimensional spaces and it has to map the spread of

a population into a statistically reasonable indicator value. The meaning and calculation is quite intuitive: ASM determines the Euclidean distance of all possible combinations of solutions without repetition and without taking the point order into account:

$$\Delta = \sum_{i,j=1\ldots n, i<j} \frac{|d_{ij} - \bar{d}|}{N} \qquad \text{with } N = \frac{n!}{2!(n-2)!} = \binom{n}{2}, \tag{5.24}$$

where d_{ij} symbolizes the Euclidean distance between the solutions i and j, n is the total number of solutions, $\bar{d}$ the average distance over all determined distances and N is the number of calculated distances. More precisely, N is the number of possible considerations of two objects from a set of n objects. The computational complexity of ASM for a solution set of n individuals and k objectives is composed of n^2 subtractions, $n^2 + 1$ operations for $\bar{d}$ and a complexity of $O(n^2 \cdot k)$ for the calculation of all Euclidean distances d_{ij}. Therefore, the worst computational complexity for ASM is $O(n^2 \cdot k)$. ASM is used as diversity metric in the experiments of this thesis and is always denoted as diversity.

5.3 Statistical Evaluation

The experimental results of the landscape analysis and the test series of different VONSEA configurations are evaluated by descriptive methods: Determination of the location parameters and of the distribution of the numerical data. For a suitable graphical representation, boxplots are created of the different test series to provide an overview of the average and spread of the numerical data at one glance. Boxplots are commonly used to graphically present a numerical data set, since they provide a simple insight into data symmetry and skewness. Boxplots or Box-Whisker-Plots are a five-point-summarization of the numerical data. The characteristic five points of a boxplot are the minimum x_{min}, the 25%-quartile, the median, the 75%-quartile and the maximum x_{max} of the numerical data. The box labels 50% of the data limited by the first and third quartile. The length of the box is determined by the 25% and 75% quartile and is termed inter-quartile range. The median is depicted by the additional line in the box. The whiskers represent x_{min} and x_{max} of the data. The length of the whiskers is not more than 1.5-times of the inter-quartile range. Otherwise, minimal or maximal values which differ more than 1.5-times from the inter-quartile range are outliers and presented as dots in the figures. [99]

The interpretation of the experimental results of the landscape analysis assumes a comparison of the boxplots to formulate some hypothesis regarding the design of VONSEA. This comparison is performed by the determination of the average increase or decrease of the five-point boxplot values (x_{min}, 25%-quartile, median, 75%-quartile and x_{max}). The geometric mean is the most common average value to determine the central tendency of an increase or decrease of a data set. It is defined via the n-th root of the product of n numbers

$$\sqrt[n]{x_1 \cdot x_2 \cdots x_n}, \tag{5.25}$$

where $n = 5$ is the number of the boxplot values and $x_1, x_2, ..., x_5$ are the five percentage deviations of these five values [61]. The boxplots depicting the VONSEA performance present averaged five-point boxplot values of the total number of test runs for each configuration. The significance of the averaged five-point boxplot values is verified on a two-sampled t-test [147]. These averaged five-point value sets are divided into two equally sized samples and the t-test is performed on these two samples. The t-test is a hypothesis test that analyze if the mean difference of two sample means is a consequence of the sample distributions. The two-sample t-test for unpaired data[1] is performed. The precondition of a t-test is that the numerical data follow a Gaussian normal distribution.
The first step of the t-test is the formulation of the null hypothesis (H_0) and the alternative hypothesis (H_1):

$$H_0 : \ \mu_1 = \mu_2, \ H_1 : \ \mu_1 \neq \mu_2. \tag{5.26}$$

The second step is the calculation of the test value t.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{(\bar{x}_1 - \bar{x}_2)}}, \tag{5.27}$$

where $\bar{x}_1, \bar{x}_2$ are the sample means and $\sigma_{(\bar{x}_1 - \bar{x}_2)}$ is the estimated standard error of the mean difference in the data set. The calculation of the standard error depends on whether the samples have a variance homogeneity. The sample variances are homogeneous in the case that there is no significant difference between the sample variances, otherwise the sample variance is termed heterogeneous. In the case of homogeneous sample variances, the standard error is calculated by:

$$\sigma_{(\bar{x}_1 - \bar{x}_2)} = \sqrt{\frac{(N_1 - 1) \cdot s_1^2 + (N_2 - 1) \cdot s_2^2}{(N_1 - 1) + (N_2 - 1)}} \cdot \sqrt{\frac{1}{N_1} + \frac{1}{N_2}}, \tag{5.28}$$

[1] Unpaired data have no affecting connection to each other. [99]

where N_1, N_2 are the sample sizes and s_1^2, s_2^2 are the sample variances. In the case of heterogeneous sample variances, the standard error is determined by:

$$\sigma_{(\bar{x}_1-\bar{x}_2)} = \sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}} \tag{5.29}$$

The third step is the decision if the null hypothesis has to be rejected. Therefore, the probability for the test value t and the critical value of the t-distribution have to be determined. The null hypothesis H_0 is then rejected if $|t| > t_{1-\alpha/2,\nu}$, where $t_{1-\alpha/2,\nu}$ with $\nu = N_1 + N_2 - 2$ is the critical value of the t-distribution. The significance level is chosen as $\alpha = 0.05$. [99]
Result matrices are given as a summarization of the experimental results in form of confidence limits for the mean. The 95%-confidence limits for the mean are defined by:

$$\bar{Y} \pm t_{a-\alpha/2,N-1}\frac{s}{\sqrt{N}}, \tag{5.30}$$

where $\bar{Y}$ is the sample mean, s is the sample standard deviation, N is the sample size, α is the significance level and $t_{1-\alpha/2,N-1}$ is the percentile of the t-distribution with $N-1$ degrees of freedom. [147]

6 Experimental Results

In this chapter, the VONSEA components are systematically investigated on a 3D-MOP consisting of the molecular fitness functions NMW, MW and Hydro as well as on a 4D-MOP comprising NMW, MW, Hydro and InstInd. These fitness functions act comparatively in the way that candidate solutions are compared to a pre-defined reference solution (eq. (3.23)). Therefore, the fitness functions have to be minimized. Only positive fitness values are possible due to the use of the absolute value on the differences between the fitness values of the pre-defined reference and the candidate solution.
The experiments are evaluated by the ACV, ACV_{scaled}, ACV_{rel} and diversity indicator. The ACV and ACV_{scaled} (eq. (5.10) and eq. (5.11)) are applied as the convergence indicators for the entire populations with the zero point as pre-defined ideal point for both MOPs. The diversity indicator ASM (eq. (5.24)) is applied to measure the solution distribution for the entire population and is further denoted as diversity. ACV_{rel} (eq. (5.12))is applied to analyze the relative non-dominated solution quality.
Each configuration is run 30 times until the 18th generation, and every generation of each run is evaluated according to the four indicators. The boxplots presented are created according to the descriptions in Section 5.3. A very large number of configurations have been examined, especially of the different component parameters, but to achieve a clear and structured overview which demonstrates the essential messages, the most meaningful indicators were used to choose the necessary number of configuration results for representation.
Firstly, the variation operators are investigated regarding an improved VONSEA performance. Afterwards, the selection strategies are examined for a further improvement of the performance. Then, the interaction between population size and selection is in the focus. The last series of tests is concerned with the direct comparison of VONSEA to the traditional NSGA-II in terms of the non-dominated solutions in the first generations.

	Recombinations			
Mutations	*Random*	*LiDeRP*	*ExpoDeRP*	*2-point-edges*
Random	RanRan	RanLin	RanExpo	RanEdges
AAWeighted	AAWeiRan	AAWeiLin	AAWeiExpo	AAWeiEdges
BSself	BSselfRan	BSselfLin	BSselfExpo	BSselfEdges
BS_{adap}	BSRan	BSLin	BSExpo	BSEdges
LiDeMut	LinRan	LinLin	LinExpo	LinEdges
QuadDeMu	QuadRan	QuadLin	Quad Expo	QuadEdges
Thierens	CGainRan	CGainLin	CGainExpo	CGainEdges

Table 6.1: Overview of all possible variation operator combinations and their denotation

6.1 Recombination and Mutation

The following test series investigate the interaction of the different recombination and mutation operators with regard to their characteristic interaction behavior. Table 6.1 gives an overview of the investigated configurations with various variation operators and their abbreviations that are used in the following. These configurations make use of the default selection strategy, the default population size and the default number of parents for recombination: The aggregate selection is applied with the parameters $p_0 = 50\%$ and $ts = 10$ as well as a population size of 100 and three parents for recombination. The main goal of these test series is to find a recombination-mutation combination providing constantly good performance for both MOPs. Ideally, this combination has a high potential to provide remarkably good performance results for a wide range of similar multi-objective biochemical optimization problems similar to those proposed in this thesis. Some of these operators are parameter-dependent and a fine-tuning is advisable to improve performance.

6.1.1 Random Recombination with Mutation Variation

6.1.1.1 Experiments with Configuration RanRan

The first test series examine the performance of the configuration RanRan. As these random operators have two parameters, the actuarial expectation μ and the standard deviation σ, different parameter settings have to be investigated for μ and σ. Figure 6.1 and 6.2 depict the convergence and diversity performance of configuration RanRan in the case of the 3D-MOP with a mutation

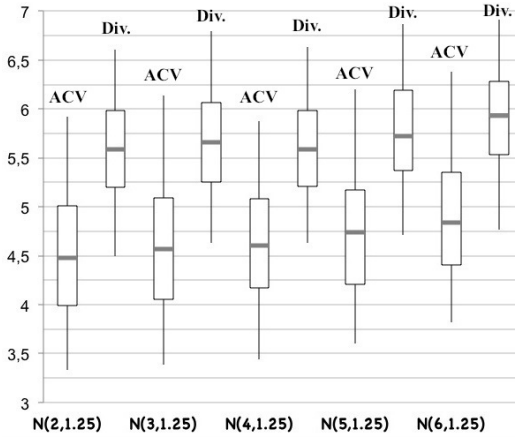

Fig. 6.1: Performance of configuration RanRan in the case of 3D-MOP with a variation of the mutation parameter μ and $\sigma = 1.25$.

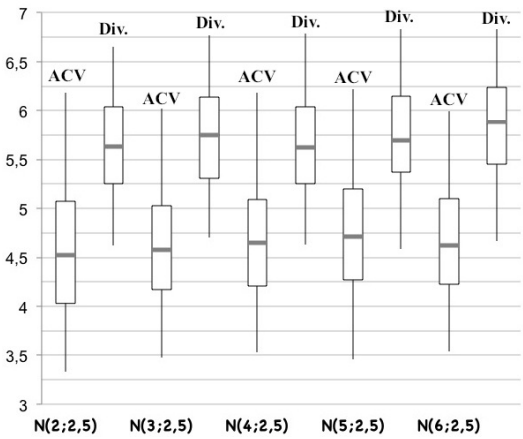

Fig. 6.2: Performance of configuration RanRan in the case of 3D-MOP with a variation of the mutation parameter μ and $\sigma = 2.5$.

parameter variation of μ and σ, denoted by $N(\mu, \sigma)$. The recombination parameter are fixed with $\mu = 2$ and $\sigma = 2.5$ and a variation of these parameters are subject of investigations in a later part of this paragraph. Figure 6.1 presents the performance results of the actuarial expectation variation with $\sigma = 1.25$. In general, the increase of μ results in an increase of the ACV and the diversity values and therefore in worse convergence behavior with an improvement of the diversity at the same time. $N(2, 1.25)$ is the preferable choice with regard to a good balance between convergence and diversity performance as this configuration achieves the lowest ACV values and acceptable diversity values. An increase of the diversity values correlates to an increase of the ACV values. Figure 6.2 depicts the performance results for $\sigma = 2.5$ and a variation of μ. In general, the ACV and diversity values are more balanced to each other compared to Figure 6.1. The ACV values reveal a slight increase by the increase of μ up to 5. $N(6, 2.5)$ is nearly of the same convergence level like $N(3, 2.5)$. The diversity values reveal an oscillating behavior by the increase of μ. The preferable parameter setting with regard to a suitable convergence-diversity balance is $N(3, 2.5)$, which is comparable to $N(4, 1.25)$ in performance. However, the inter-quartile range of $N(3, 2.5)$ is higher compared to $N(4, 1.25)$ whose are the default settings of the random mutation. A further increase of σ is not advisable as the performance values are becoming more and more balanced. This is a potential consequence of the flatter shape of the probability density function (Figure 4.14). A decrease of the σ values results in nearly constant mutation rates, visible by the shape of the probability function.

Figure 6.3 and 6.4 present the performance of the configuration RanRan achie-

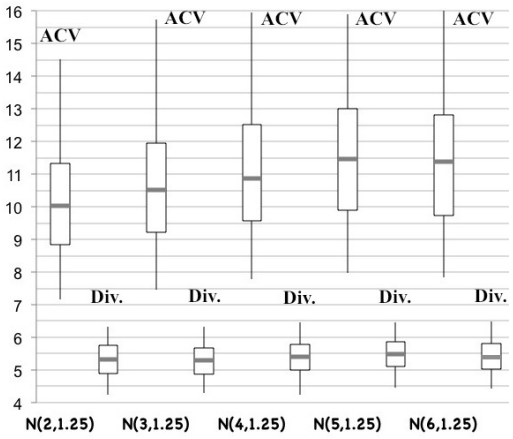

Fig. 6.3: Performance of configuration RanRan in the case of 4D-MOP with a variation of mutation parameter μ and $\sigma = 1.25$.

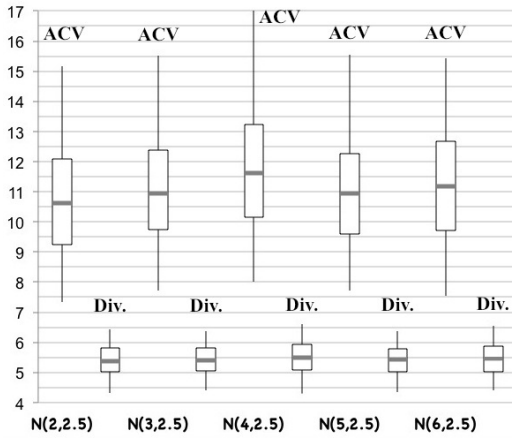

Fig. 6.4: Performance of configuration RanRan in the case of 4D-MOP with a variation of mutation parameter μ and $\sigma = 2.5$.

ved in the case of the 4D-MOP. The variation of σ generally reveals a higher influence on the convergence behavior than on diversity. The diversity values (Figure 6.3, 6.4) are oscillating by an increase of μ for $\sigma = 1.25$ and 2.5. The convergence is slowed down by an increase of μ presented by the ACV values up to $\mu = 5$ (Figure 6.3) and up to $\mu = 4$ (upper boxplots in Figure 6.4). The preferable parameter settings regarding a good convergence-diversity balance are given by $N(2, 1.25)$. The ACV values for $\sigma = 2.5$ are generally higher than for $\sigma = 1.25$, whereas the diversity values are comparable.
These first experiments of the mutation parameter variation with the 3D- and 4D-MOP reveal that a small shape of the probability density function with $\sigma = 1.25$ is generally more advisable than higher values. Otherwise, a lower number of μ is more advisable in the case of an increase of the problem dimension. As a consequence, μ has to be adapted for each optimization problem making random mutation quite unattractive.

The second test series examine the influence of random recombination parameter variation on the VONSEA performance in the case of the 3D- and 4D-MOP. Figure 6.5 presents the 3D-MOP performance results of configuration RanRan with fixed mutation parameters of $N(4, 1.25)$ and a variation of the recombination parameters μ, σ. The mutation settings $N(4, 1.25)$ are default settings for the random mutation and used within these test series instead of the preferable settings identified in the previous test runs. As there is an interdependence between the parameters μ and σ of the random recombination and mutation operator, a fine-tuning of one random operator requires fixed parameter settings of the other. An increase of μ results in an increase of the

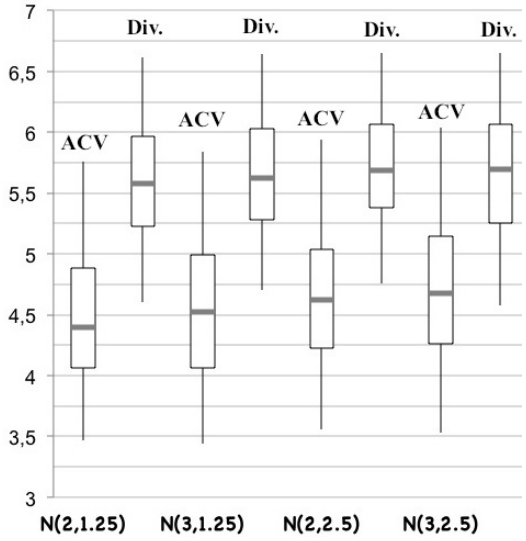

Fig. 6.5: Performance of configuration RanRan with a variation of recombination parameter μ, σ with mutation parameter settings of $N(4, 1.25)$. (3D-MOP)

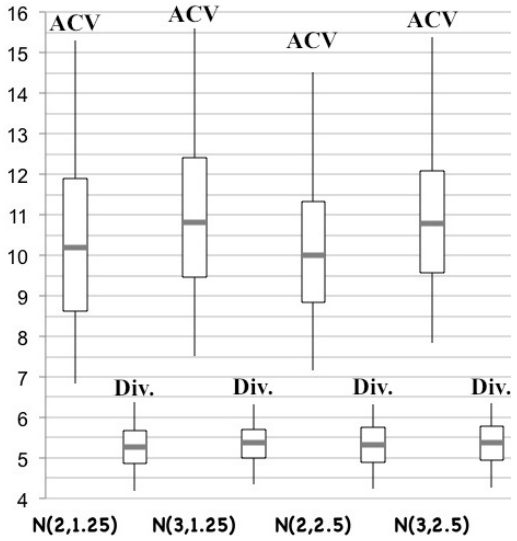

Fig. 6.6: Performance of configuration RanRan with a variation of recombination parameter μ, σ with mutation parameter settings of $N(2, 1.25)$. (4D-MOP)

ACV values independent of σ. This is a consequence of the fact that a significantly higher number of recombination points results in a higher number of disruptions of potentially high qualified genetic material. The highest diversity values are achieved for $\sigma = 2.5$ and are nearly constant by an increase of μ. Otherwise, the diversity values increase by an increase of μ for $\sigma = 1.25$. The preferable settings for good convergence-diversity balance are $N(2, 2.5)$ with the highest diversity values and slightly increased ACV values compared to $N(3, 1.25)$.

Figure 6.6 depicts the 4D-MOP performance results of RanRan with the fixed mutation parameter of $N(2, 1.25)$ and variable recombination parameters. Once more, an increase of μ results in an increase of the ACV values (Figure 6.6) for $\sigma = 1.25$ and $\sigma = 12.5$. The highest diversity values (lower boxplots of Figure 6.6) are achieved for $\sigma = 2.5$. The preferable parameter settings for a good convergence-diversity balance are $N(2, 2.5)$ with comparable and constant low ACV values.

The second test series of recombination parameter variation with the 3D- and 4D-MOP advises the same preferable parameter settings for both optimization problems. However, these results are not generalizable without further experiments with several others optimization problems. A remarkable influence of the recombination parameters is clearly visible and makes the use of the random recombination operators as a black-box configuration operator for other

similar multi-objective biochemical optimization problems unattractive.

6.1.1.2 Experiments with Configuration AAWeiRan

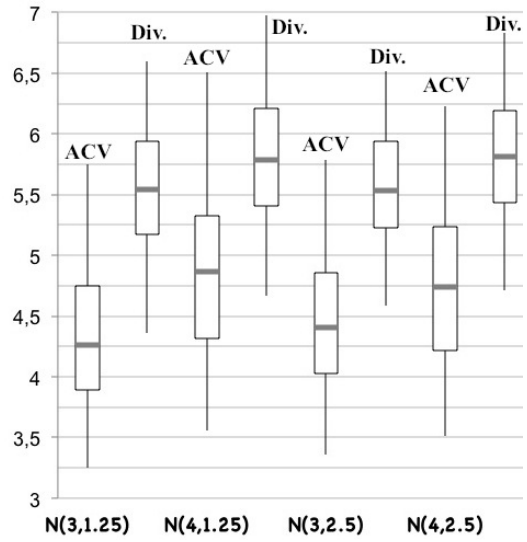

Fig. 6.7: Performance of configuration AAWeiRan with varying mutation parameter parameter μ, σ and fixed recombination parameter settings of $N(2, 2.5)$. (3D-MOP)

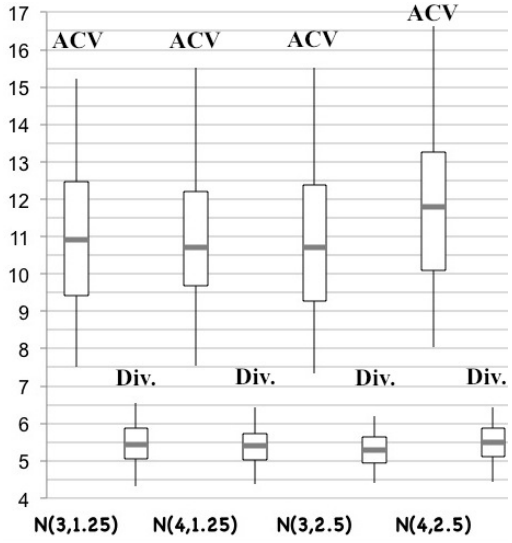

Fig. 6.8: Performance of configuration AAWeiRan with varying mutation parameter parameter μ, σ and fixed recombination parameter settings of $N(2, 2.5)$. (4D-MOP)

These test series investigate the performance of configuration AAWeiRan in the case of the 3D- and 4D-MOP. Mutation operator AAWeighted is similar to random mutation and has therefore the same parameters μ and σ, but the mutated amino acids have a specific frequency to be mutated. The frequencies are equal to the natural incidence of each amino acid. Figure 6.7 presents the 3D-MOP performance results of AAWeiRan with different parameters μ, σ. An increase of μ results in an increase of the ACV values as well as the diversity values, independent of σ. Once more, there is an interdependence between the parameter settings of μ, σ of the mutation and recombination operator. A fine-tuning of the mutation parameters requires fixed recombination parameters, which are set to $N(2, 2.5)$. The preferable settings for good convergence-diversity balance are $N(4, 2.5)$. The comparison of these results to those of RanRan with preferable parameter settings ($N(2, 1.25)$ in the case of mutation, $N(2.2.5)$ in the case of recombination) reveals that the ACV values and the diversity values are both slightly decreased compared to RanRan (Figure 6.1). Consequently, there is no advantage in general of AAWeighted mutation compared to random mutation.

Figure 6.8 depicts the 4D-MOP performance results of AAWeiRan with fixed recombination parameters $N(2.2.5)$. In general, there is no noticeable effect on the performances caused by a variation of the mutation parameter μ, σ. The preferable parameters for good convergence-diversity balance and constant results - specified by the lower inter-quartile range - is $N(4, 1.25)$. The comparison of these results to those of RanRan with preferable parameter setting ($N(2, 1.25)$ in the case of mutation $N(2.2.5)$ in the case of recombination) reveals that the ACV values are on average remarkably higher in the case of AAWeiRan compared to RanRan (Figure 6.3). The diversity values are on average slightly higher in return. Once more, there is no remarkable benefit of configuration AAWeiRan over RanRan.

6.1.1.3 Experiments of the Configuration BSselfRan

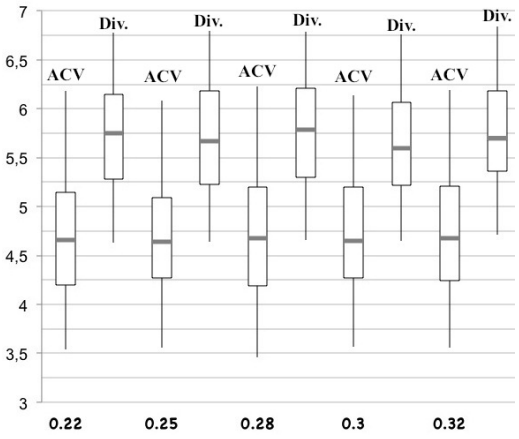

Fig. 6.9: Performance of configuration BSselfRan with variation of learning rate γ and fixed recombination parameters $N(2, 2.5)$. (3D-MOP)

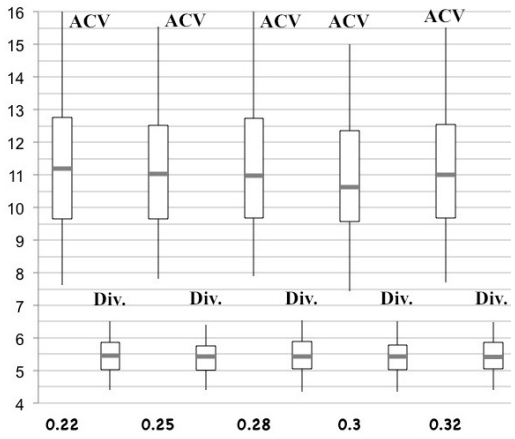

Fig. 6.10: Performance of configuration BSselfRan with variation of learning rate γ and fixed recombination parameters $N(2, 2.5)$. (4D-MOP)

The following test series examine the 3D- and 4D-MOP performance results of configuration BSselfRan. Mutation BSselfadaptive has a start mutation rate of 0.2, which is the default start mutation probability of the mutation operators in VONSEA except in the case of mutations LiDeMut and QuadDeMut. Figure 6.9 presents the 3D- performance results of BSselfRan with fixed recombination parameters $N(2, 2, 5)$ and different values for the learning rate γ. In general, there is no significant influence of the γ values, neither on the diversity nor on the convergence. Compared to configuration RanRan, the diversity values are on the same level, but the ACV values are in general remarkablly higher in the case of BSselfRan compared to RanRan (Figure 6.5). This allows the

hypothesis that the random recombination is mainly responsible for the solutions diversity. Furthermore, the learning rate γ is not able to use its influence on the performance results.
Figure 6.10 depicts the 4D-performance results of BSselfRan with different γ values and fixed recombination parameters $N(2, 2.5)$. Once again, the variation of γ reveals no visible regularity. The diversity level is comparable to RanRan (Figure 6.6), but the convergence is significantly slowed down in the case of BSselfRan compared to RanRan. Further experiments with the aim of performance improvement by an adaption of the recombination parameters are not reasonable, since these fine-tuned parameters are generally problem-oriented.

6.1.1.4 Experiments with Configurations BSRan, LinRan, QuadRan and CGainRan

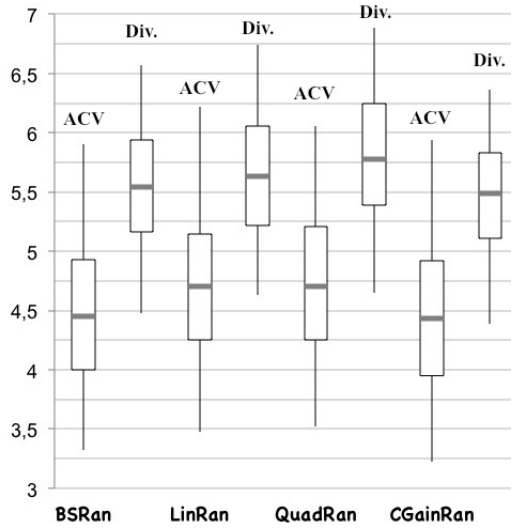

Fig. 6.11: Performance of BSRan, LinRan, QuadRan and CGainRan with fixed recombination parameters $N(2, 2.5)$. (3D-MOP)

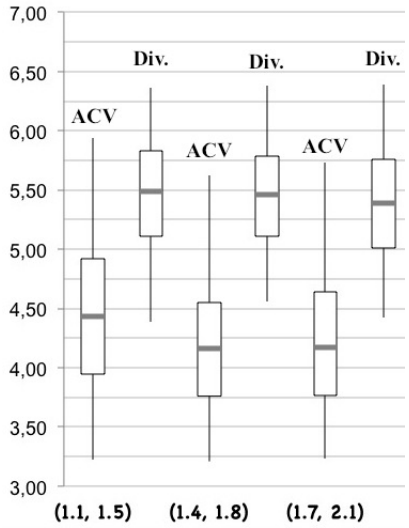

Fig. 6.12: Performance of CGainRan with different parameter settings (α, ω) and fixed recombination parameters $N(2, 2.5)$. (3D-MOP)

Figure 6.11 depicts the 3D-performance results of the configuration BSRan, LinRan, QuadRan and CGainRan with the recombination parameters $N(2, 2.5)$ and the exploration and learning factor $\omega = 1.5$, $\alpha = 1.1$. Configuration CGainRan achieves the lowest ACV values and therefore the best convergence performance, but the diversity is significantly lower. The convergence behavior of configuration LinRan and QuadRan are comparable and reveal the highest ACV values compared to BSRan and CGainRan. QuadRan achieves the highest diversity values as mutation QuadDeMut provides constant high mutation

probabilities in the first generations. BSRan achieves the lowest ACV values with slightly decreased diversity values. As the convergence improvement is significantly higher than the decrease of the diversity values compared to LinRan, BSRan is regarded as the preferable configuration. BSRan, is further comparable in performance with AAWeiRan (Figure 6.7) and RanRan (Figure 6.1) with mutation parameters $N(3, 1.25)$ and $N(2, 1.25)$ respectively.
Figure 6.12 presents the 3D-performance results of CGainRan with different parameter settings. In general, CGainRan achieves the best convergence behavior compared to the other configurations, but reveals also the lowest diversity. This allows the conclusion that the use of the mutation probability according to the lowest ACV results in highly qualified but clustered solutions. The preferable setting of these parameters for good convergence-diversity balance is $\omega = 1.8$, $\alpha = 1.4$, since the diversity values are at most high and the ACV values are the lowest compared to the other settings.
Figure 6.13 presents the 4D-performance results of the configurations BSRan,

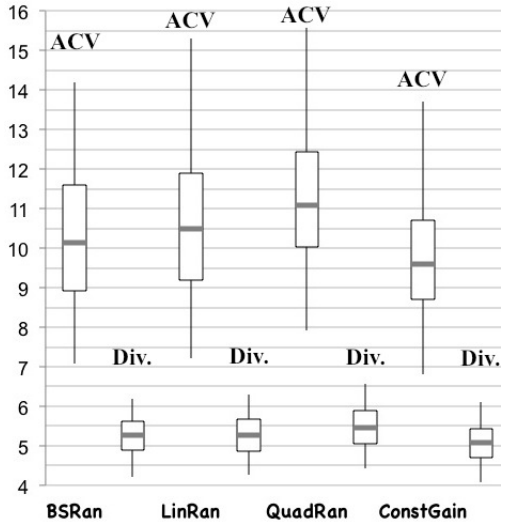

Fig. 6.13: Performance of BSRan, LinRan, QuadRan and CGainRan with fixed recombination parameters $N(2, 2.5)$. (4D-MOP)

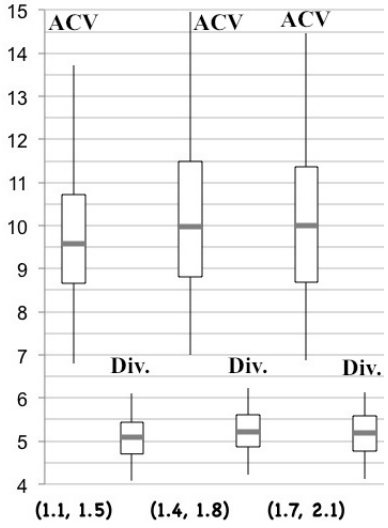

Fig. 6.14: Performance of CGainRan with different parameter settings (α, ω) and fixed recombination parameters $N(2, 2.5)$. (4D-MOP)

LinRan, QuadRan and CGainRan with the recombination settings $N(2, 2.5)$. The results are very similar to those of the 3D-performance results above in a qualitative way: QuadRan achieves the highest diversity, but also the worst convergence. CGainRan has the best convergence behavior at the cost of diversity. BSRan reveals good convergence values with only a very slight decrease of the diversity values compared to LinRan. Therefore, LinRan is regarded as preferable configuration. In general, BSRan achieves the best convergence be-

havior compared to the configurations BSselfRan (Figure 6.10) and AAWeiRan 6.8 with comparable diversity values. BSRan has comparable performance to RanRan (Figure 6.3) with the advisable mutation parameters $N(2, 1.25)$.
Figure 6.14 depicts the 4D-performance configurations of CGainRan with different parameter settings. Once more, CGainRan achieves the best convergence behavior of all configurations so far at the cost of diversity. The preferable parameter settings for a good convergence-diversity balance are $\omega = 2.1$, $\alpha = 1.7$ and therefore different to the preferable settings in the case of the 3D-MOP.

6.1.2 LiDeRP with Mutation Variation

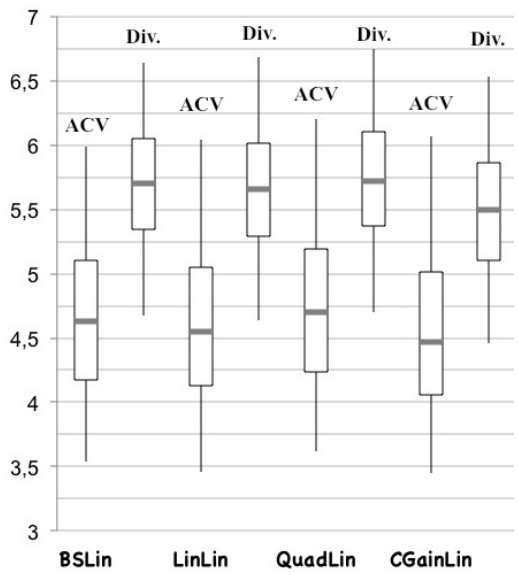

Fig. 6.15: Performance of BSLin, LinLin, QuadLin and CGainLin, (3D-MOP)

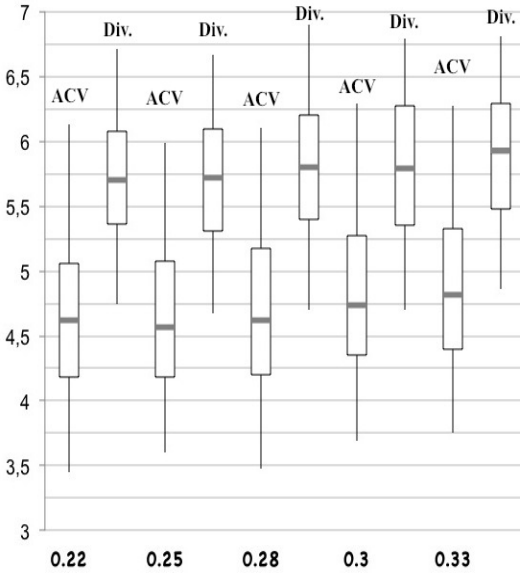

Fig. 6.16: Performance of BSselfLin with different parameter settings γ, (3D-MOP)

The following series of tests comprise the configurations of the linear recombination LiDeRP with various mutation operators. The presented results are additionally discussed with regard to a performance comparison of these configurations to Random recombination.
Figure 6.15 depicts the 3D-performance results of the configurations BSLin, LinLin, QuadLin and CGainLin (with the default setting $(\alpha, \omega) = (1.1, 1.5)$). CGain achieves once again the lowest diversity as well as ACV values, whereas QuadLin achieves the highest diversity and ACV values. BSLin and LinLin achieve comparable results for diversity and convergence. These results are compared to 3D-performance results of BSRan, LinRan, QuadRan and

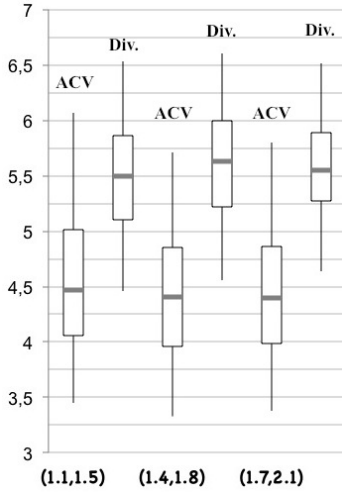

Fig. 6.17: Results of CGainLin with different parameter settings (α, ω), (3D-MOP)

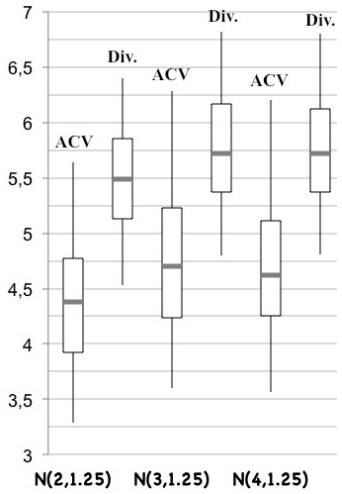

Fig. 6.18: Results of RanLin with different parameter settings (μ, σ), (3D-MOP)

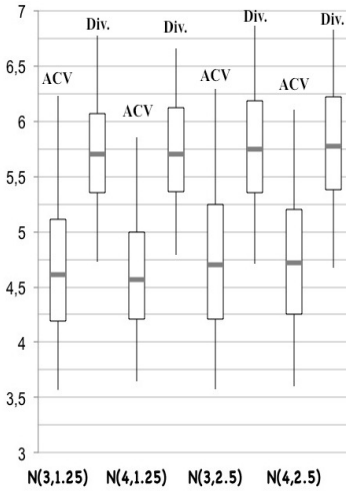

Fig. 6.19: Results of AAWeiLin with different parameter settings (μ, σ), (3D-MOP)

CGainRan (Figure 6.11) to gain an insight into the influence of the recombination operator. In general, the recombination operators in combination with these mutation operators have no significant influence on the performance results. The general tendency according to diversity and convergence are similar for both recombinations. Although, the general performance differences of the configurations are lower in the case of LiDeRP. Figure 6.17 presents the performance results of the configuration CGainLin with different parameter settings. The parameter settings $(1.4, 1.8)$ are preferable, since the ACV values have a tendency towards lower values and the diversity values are higher compared to the other settings. The results are highly comparable to those of CGainRan (Figure 6.12).

Figure 6.16 depicts the 3D-performance results of configuration BSselfLin with different parameter settings of the learning factor. The learning factor $\gamma = 0.22$ and $\gamma = 0.25$ achieve comparable results. A further increase of the learning factor reveals an increase of the ACV values and therefore a slowing down of the convergence with only a slight increase of the diversity values for the highest value $\gamma = 0.32$. Therefore, a lower learning factor is more advisable, since the performance results are more consistent for these values. The comparison of these results to BSselfRan (Figure 6.9) reveals highly comparable results. Once again, the recombination operator in combination with mutation BSself has no visible influence on the performance results.

Figure 6.18 and Figure 6.19 present the performance results of the random mutation operators Random and AAWeighted with different parameter settings.

RanLin with $N(2, 1.25)$ achieves the lowest ACV and diversity values of all configurations. An increase of μ reveals significantly higher ACV and diversity values. The results of $N(3, 1.25)$ and $N(4, 1.25)$ are comparable. Higher μ values are advisable with regard to a good convergence-diversity balance. Higher values of σ are not suitable, since the performance results are becoming more and more unstable. The performance results of configuration AAWeiLin with different parameter settings (Figure 6.19) are in general comparable to the other configurations, although higher σ-values provide a higher spread of the performance values. The comparison of the RanLin performance to RanRan (Figure 6.1) reveals comparable results except for the settings $N(3, 1.25)$. RanLin provides comparable diversity values but remarkable improved convergence values. The comparison of AAWeiLin with AAWeiRan (Figure 6.7) reveals a quite different performance: The configurations of AAWeiLin with different parameter settings are very similar to one another in contrast to AAWeiRan. The parameters of AAWeighted mutation have less influence on the performance in combination with LiDeRP.
The comparison of Figure 6.15 - Figure 6.19 reveals that BSselfLin with high parameter values provide the highest diversity as well as the highest ACV values. The configuration of CGainLin provides the lowest diversity and ACV values. The advisable configuration with regard to a good convergence-diversity balance is provided by the configurations BSLin and LinLin.
Figure 6.20 - Figure 6.24 present the 4D-performance results of the configura-

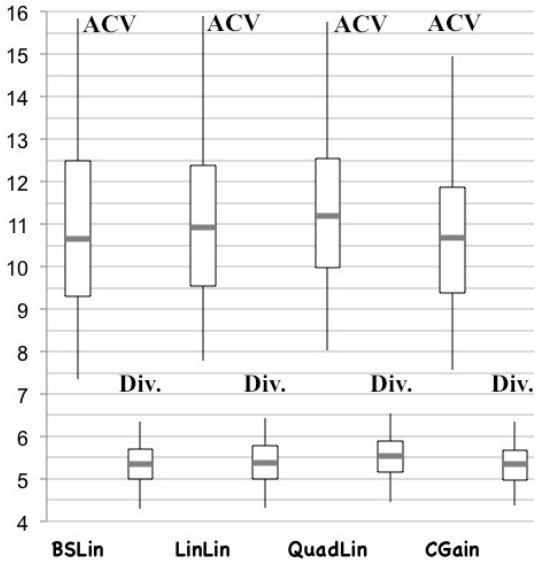

Fig. 6.20: Performance of BSLin, LinLin, QuadLin and CGainLin, (4D-MOP)

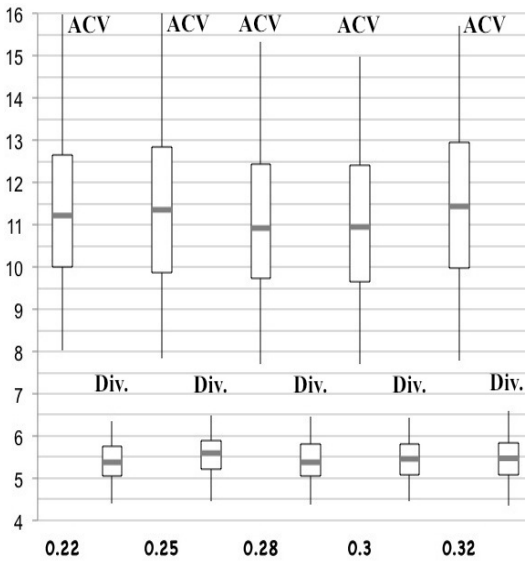

Fig. 6.21: Performance of BSselfLin with different parameter settings γ, (4D-MOP)

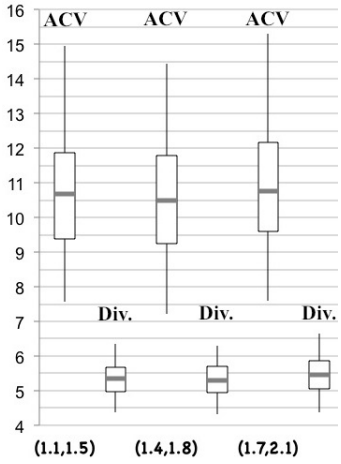

Fig. 6.22: Results of CGain-Lin with different parameter settings (α, ω), (4D-MOP)

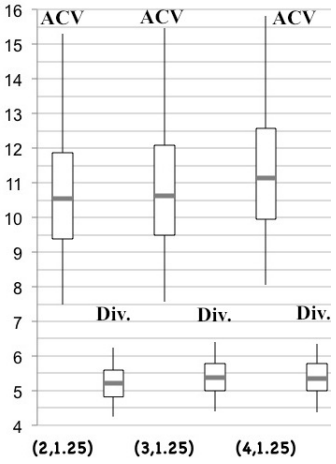

Fig. 6.23: Results of Ran-Lin with different parameter settings (μ, σ), (4D-MOP)

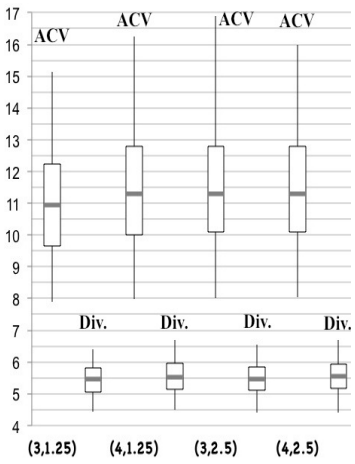

Fig. 6.24: Results of AAWei-Lin with different parameter settings (μ, σ), (4D-MOP)

tions BSLin, BSselfLin, LinLin, QuadLin, CGainLin, RanLin and AAWeiLin. The upper boxplots depict the spread of the ACV and the lower ones the spread of the diversity values. Figure 6.20 depicts the results of the configurations BS-Lin, LinLin, Quadlin and CGainLin with $(\alpha, \omega) = (1.1, 1.5)$. The performance of QuadLin has a tendency towards the highest diversity and ACV values, whereas the performance of CGainLin achieves the lowest diversity and ACV values (Figure 6.20). The performance results of BSLin and LinLin are comparable. These results have also been observed in the case of the 3D- performance results of the corresponding configurations (Figure 6.15). Furthermore, the comparison of these results to BSRan, LinRan, QuadRan and CGainRan in the case of the 4D-MOP (Figure 6.13) reveals similar performance results, although the performance differences in the case of the configurations with LiDeRP are remarkablly smaller than those of the configurations with Random recombination. The performance results of configurations BSselfLin with different learning rates γ are quite undifferentiated (Figure 6.21). The configuration with the learning rate $\gamma = 0.25$ achieves the highest diversity as well as ACV values. The configurations with $\gamma = 0.28$ and $\gamma = 0.3$ achieve the lowest diversity and ACV values. This undifferentiated performance behavior is comparable with the configurations BSselfRan with the corresponding learning rates (Figure 6.10). The configurations of CGainLin with different parameter settings provide very low metric values. The performance differences between these settings are very small, nevertheless, the settings $(\alpha, \omega) = (1.4, 1.8)$ tend to have lowest ACV values and comparable diversity values. In general, the lowest ACV and diversity values are achieved by configuration RanRan with $N(2, 1.25)$.

An increase of μ results in a slight increase of the ACV values and higher diversity values. $N(3, 1.25)$ are advisable settings with lower ACV and higher diversity values compared to the other settings. Once again, the performance differences are smaller than in the case of RanRan with the corresponding parameter settings (Figure 6.4) or in the case of the 3D-performance results of RanLin (Figure 6.18). The configurations AAWeiLin (Figure 6.24) with different parameter settings generally achieve higher metric values compared to RanLin. The settings $N(3, 1.25)$ achieve the lowest metric values compared to the other (μ, σ) settings in Figure 6.24. An increase of μ results in an increase of the metric values. An increase of σ reveals comparable performance results to $N(4, 1.25)$. Once again, the performance differences of AAWeiLin are small compared to AAWeiRan (Figure 6.8). Regarding a good convergence-diversity balance, $N(4, 1.25)$ is regarded as advisable settings for the configuration AAWeiRan. The comparison of Figure 6.20 - Figure 6.24 reveals that the configuration BSselfLin with different parameter settings generally achieves the highest metric values, whereas CGainLin as well as RanLin with $N(3, 1.25)$ achieve the lowest metric values. The configurations BSLin and LinLin are regarded as preferable configurations for a good convergence-diversity balance.

6.1.3 ExpoDeRP with Mutation Variation

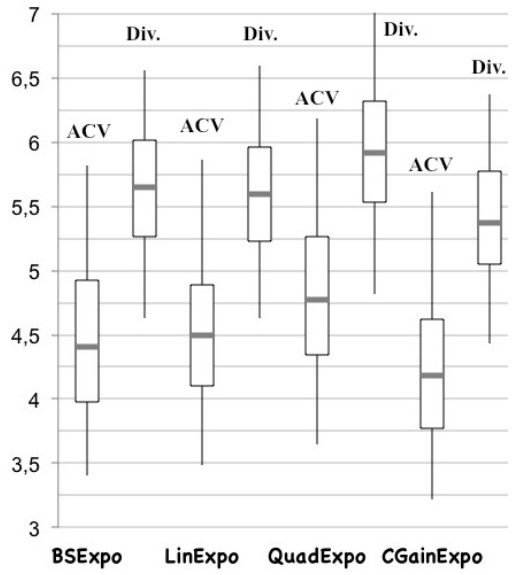

Fig. 6.25: Performance of BSExpo, LinExpo, QuadExpo and CGainExpo, (3D-MOP)

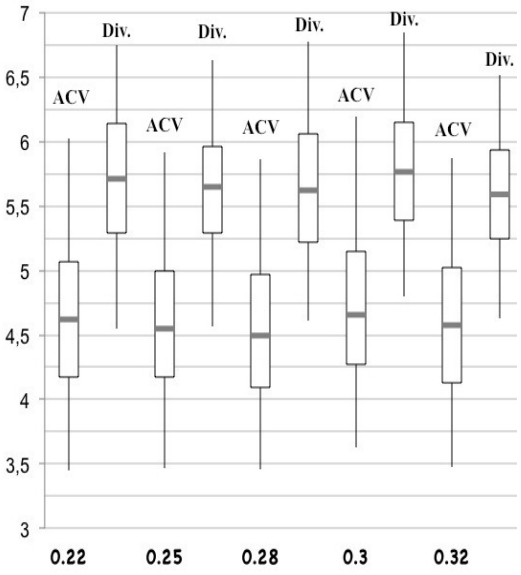

Fig. 6.26: Performance of BSselfExpo with different parameter settings γ, (3D-MOP)

These test series comprise the configurations of recombination ExpoDeRP with

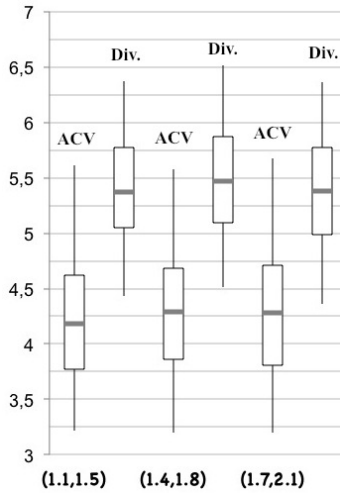

Fig. 6.27: Results of CGainExpo with different parameter settings (α, ω), (3D-MOP)

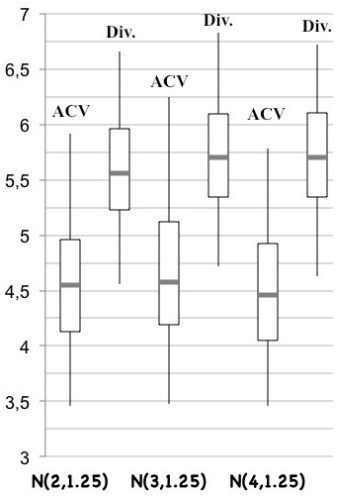

Fig. 6.28: Results of RanExpo with different parameter settings (μ, σ), (3D-MOP)

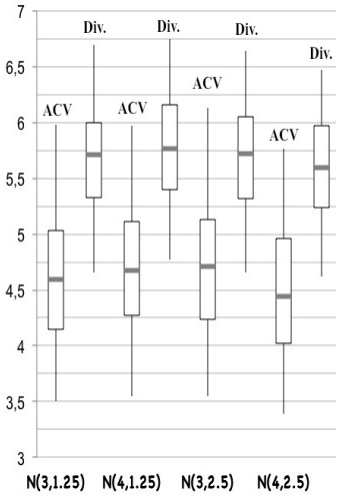

Fig. 6.29: Results of AAWeiExpo with different parameter settings (μ, σ), (3D-MOP)

various mutation operators. The presented results are additionally discussed regarding a comparison of these configurations with Random recombination and LiDeRP.

Figure 6.25 presents the 3D-configuration results of BSExpo, LinExpo, QuadExpo and CGainExpo with the parameters $(\alpha, \omega) = (1.1, 1.5)$. As in the cases of LiDeRP and Random, QuadExpo achieves the highest diversity values at the cost of convergence. Otherwise, CGainExpo reveals the lowest ACV values and therefore the best convergence at the cost of diversity. BSExpo and LinExpo reveal similar results compared to the corresponding configurations with the recombinations LiDeRP and Random and therefore a good balance of convergence and diversity, although the ACV values of BSExpo have a tendency towards lower ACV values as well as a tendency towards higher diversity values. Figure 6.26 depicts the 3D-configurations of BSselfExpo with different learning rates. The performance results are pretty undifferentiate in the cases of BSselfLin (Figure 6.16) and BSselfRan (Figure 6.9). The ACV values are in general higher that for BSExpo and LinExpo, but the diversity values are not increased at a corresponding level. The influence of the performance results does not reveal any regularity like in the cases of BSselfLin. Figure 6.27 depicts the 3D-performance results CGainExpo with different parameters. The settings $(\alpha, \omega) = (1.4, 1.8)$ reveal a tendency towards higher ACV values compared to the other settings, but also the highest diversity values. In general, CGainExpo achieves the lowest performance metrics values. Configuration RanExpo with different parameters reveals no regularity in terms

of the metric values by an increase of μ (Figure 6.28). The increase of $\mu = 2$ to $\mu = 3$ results in an increase of the metric values, but a further increase to $\mu = 4$ reveals lower ACV values but comparable diversity values. Similar performance is observed in the case of the configurations AAWeiExpo (Figure 6.29) with different parameters. For $\sigma = 1.25$, an increase of μ results in an increase of the metric values in general, otherwise the metric values decrease for $\sigma = 2.5$ and an increase of μ. The parameter settings $N(4, 2.5)$ achieve comparable performance compared to BSExpo, although the diversity values tends to be lower in the case of AAWeiExpo. In general, BSExpo and LinExpo are advisable configurations and their performance is on the same level as BSLin and LinLin. Otherwise, ExpoDeRP provides a high number of recombination points in the first generations that makes the performance prediction more difficult, especially in combination with parameter-dependent mutation operators.

Figure 6.30 - Figure 6.34 present the 4D-performance results of the configurati-

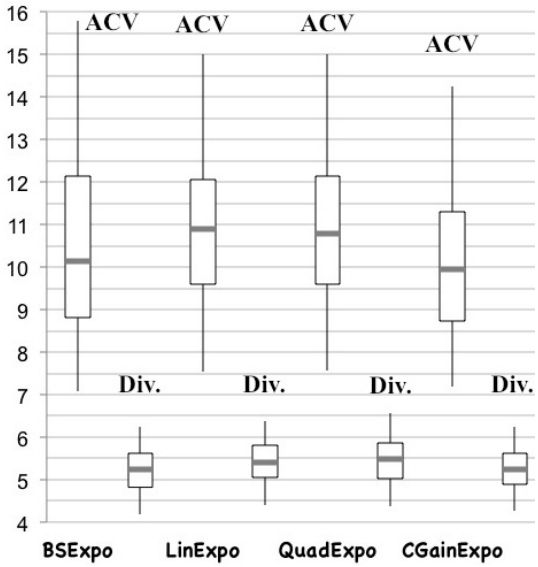

Fig. 6.30: Performance of BSExpo, LinExpo, QuadExpo and CGainExpo, (4D-MOP)

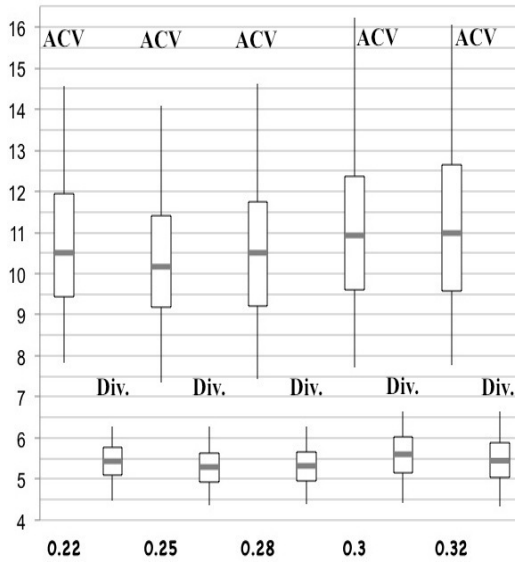

Fig. 6.31: Performance of BSselfExpo with different parameter settings γ, (4D-MOP)

ons with recombination ExpoDeRP and various mutation variants. Figure 6.30 presents the performance results of BSExpo, LinExpo, QuadExpo and CGainExpo with $(\alpha, \omega) = (1.1, 1.5)$. BSExpo has a significant tendency towards lower ACV values and therefore a better convergence performance with a tendency towards lower diversity values at the same time compared to LinExpo. As in the case of the configurations QuadLin and QuadRan, QuadExpo achieves the highest ACV and diversity values, but this tendency towards higher metric

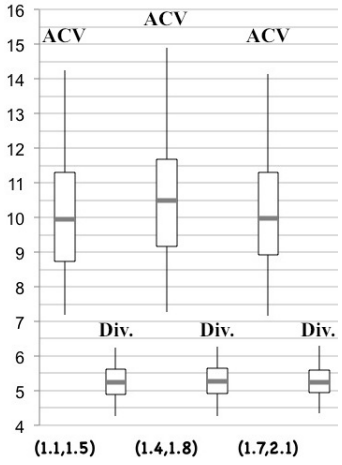

Fig. 6.32: Results of CGainExpo with different parameter settings (α, ω), (4D-MOP)

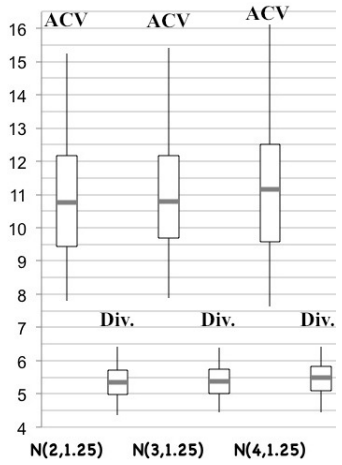

Fig. 6.33: Results of RanExpo with different parameter settings (μ, σ), (4D–MOP)

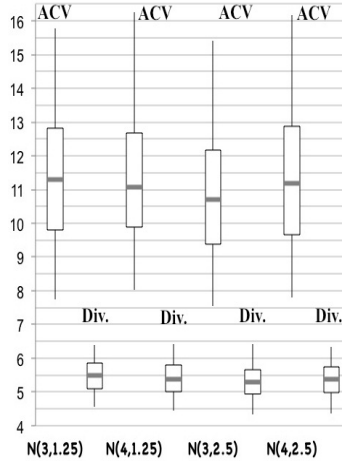

Fig. 6.34: Results of AAWeiExpo with different parameter settings (μ, σ), (4D–MOP)

values is only low. Once again, CGainExpo achieves the lowest metric values. Figure 6.31 presents the performance results of BSselfExpo with different learning rates. The configuration with $\gamma = 0.25$ achieves the lowest metric values, whereas the configuration with $\gamma = 0.3$ has the highest metric values. There is no regularity observable regarding the performance results by an increase of the learning rate. Otherwise, these performance results are comparable to the 3D-performance results of BSselfExpo (Figure 6.25). The configurations of CGainExpo with different parameter setting achieve comparable diversity values for all parameter settings (Figure 6.32). The configuration with the parameters $(\alpha, \omega) = (1.4, 1.8)$ reveals the highest ACV values and the ACV values of the other parameter settings are comparable. In general, there is no regularity observable with regard to the performance for the different parameter settings. In the case of the configurations RanExpo with different parameter settings (Figure 6.33), an increase of μ tends to result in an increase of the metric values. The lowest ACV inter-quartile range is achieved with the configuration $N(3, 1.25)$ and these parameter settings are regarded as preferable settings. Regarding the performance results of the configuration AAWeiExpo (Figure 6.34), no regularity is observable for the different parameter settings. An increase of μ for $\sigma = 1.25$ results in an increase of the metric values. Otherwise, the increase of μ for $\sigma = 2.5$ results in an decrease of the metric values. The preferable parameter settings are $N(3, 2.5)$ with low ACV values and a tendency towards higher diversity values.

The comparison of the 4D-performance results of the configurations BSExpo,

LinExpo, QuadExpo, CGainExpo and BSselfExpo are comparable to those of BSLin, LinLin, QuadLin, CGainLin and BSselfLin, although the metric values tend to be lower in the case of the configurations with ExpoDeRP. The preferable configurations regarding a good convergence-diversity balance are BSExpo and LinExpo that is comparable in its performance results to BSselfExpo with $\gamma = 0.22$.

6.1.4 2-Point-Edges with Mutation Variation

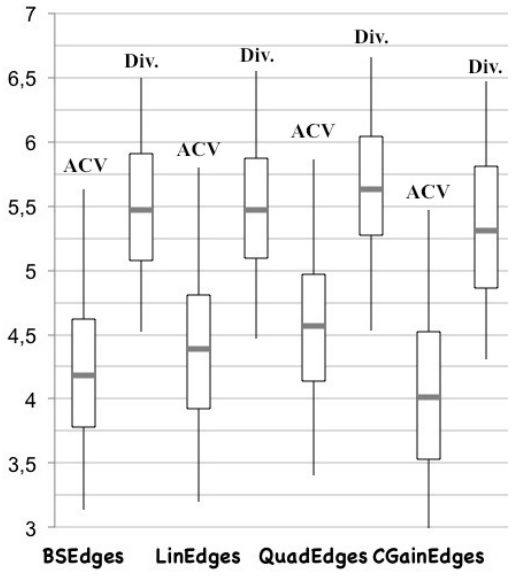

Fig. 6.35: Performance of BSEdges, LinEdges, QuadEdges and CGainEdges, (3D-MOP)

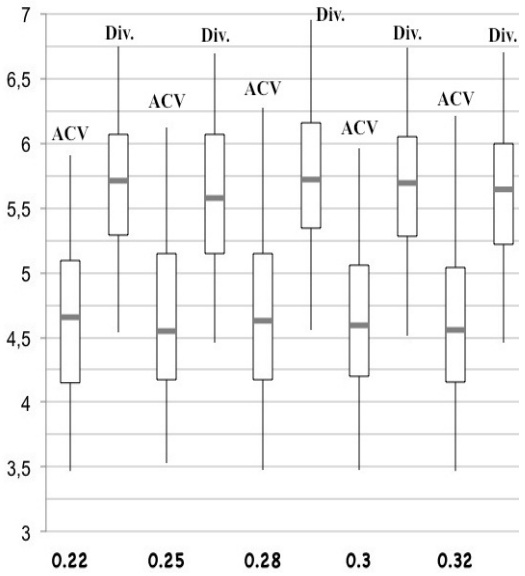

Fig. 6.36: Performance of BSselfEdges with different parameter settings γ, (3D-MOP)

The following series of tests present the performance results of the configurations with the 2-point-edges recombination and various mutation operators. The 3D-performance results of BSEdges, LinEdges, QuadEdges and CGainEdges are similar to the performance results of the corresponding configurations with Random, LiDeRP and ExpoDeRP recombination (Figure 6.11, Figure 6.15 and Figure 6.25): The configuration QuadEdges reveals the highest metric values and CGainEdges with the parameter settings $(\alpha, \omega) = (1.1, 1.5)$ the lowest metric values (Figure 6.35). BSEdges and LinEdges achieve comparable diversity values, but BSEdges reveals remarkably lower ACV values and therefore a better convergence behavior. Figure 6.36 presents the performance results of the configurations BSselfEdges with different parameter settings. Once again, the different learning rates do not reveal any regularity regarding the convergence and diversity results: The ACV values are comparable but

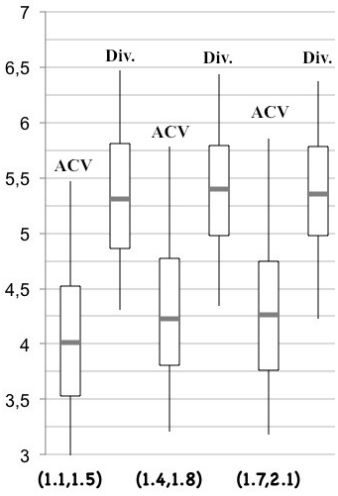

Fig. 6.37: Results of CGainEdges with different parameter settings (α, ω), (3D-MOP)

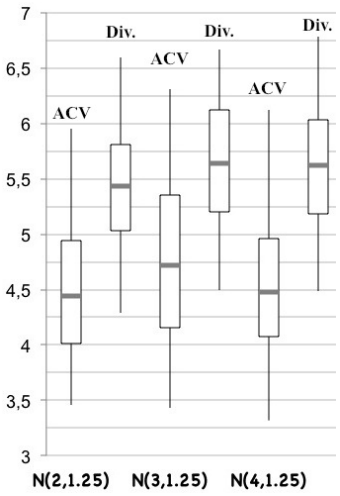

Fig. 6.38: Results of RanEdges with different parameter settings (μ, σ), (3D-MOP)

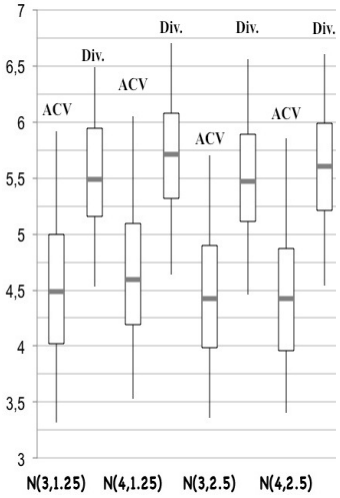

Fig. 6.39: Results of AAWeiEdges with different parameter settings (μ, σ), (3D-MOP)

reveal high inter-quartile ranges for the values $\gamma = 0.25$ and 0.28. The highest diversity values are achieved for $\gamma = 0.28$. A preferable setting for the learning rate is $\gamma = 0.2$ regarding a good convergence-diversity balance. In general, BSselfEdges achieves remarkably higher ACV and diversity values than the configurations in Figure 6.35. The increase of the parameters settings of the configuration CGainEdges achieves similar diversity values, but remarkably increased ACV values (Figure 6.37). Hence, the parameters $(\alpha, \omega) = (1.1, 1.5)$ are regarded as the preferable settings for the configuration CGainEdges achieving a good convergence-diversity balance. The increase of the parameter μ in the case of RanEdges results in an increase of the metric values in general (Figure 6.38). The parameters $N(3, 1.25)$ achieve the highest inter-quartile range of the metric values. The tendency towards higher ACV values is significantly increased compared to the other parameter settings. The parameters $N(3, 1.25)$ reveal comparable ACV values and higher diversity values than the settings $N(2, 1.25)$ and are therefore regarded as the preferable settings for the configuration RanEdges. The performance results of AAWeiEdges with different settings reveal once again no regularity according to an increase of μ or σ (Figure 6.39). An increase of μ for $\sigma = 1.25$ results in increased metric values, whereas an increase of μ for $\sigma = 2.5$ results in a decrease of the metric values. In general, the metric values of AAWeiEdges have a tendency towards higher values compared to RanEdges. The settings $N(4, 2.5)$ are regarded as the preferable settings for AAWeiEdges. From a global point of view, the configu-

rations of CGainEdges achieve the lowest metric values and the configurations QuadEdges and BSselfEdges reveal a tendency towards the highest metric values. The configurations BSEdges and LinEdges achieve a good performance regarding the convergence-diversity balance.
The performance values of all configurations with 2-point-edges recombination tend to lower metric values than those of LiDeRP and slightly lower than those of ExpoDeRP.
Figure 6.40 - Figure 6.44 present the 4D-performance results of the configurati-

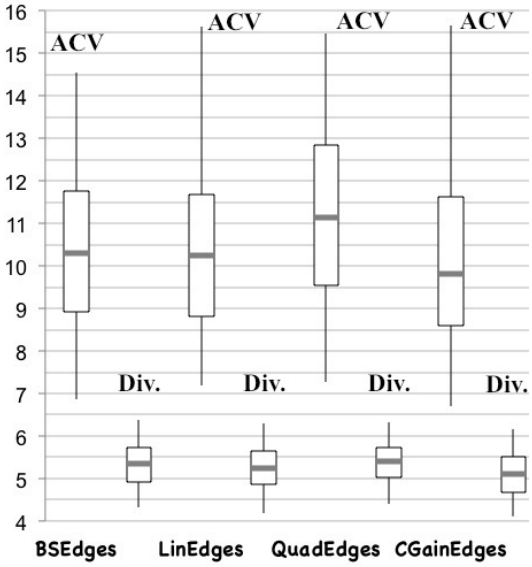

Fig. 6.40: Performance of BSEdges, LinEdges, QuadEdges and CGainEdges, (4D-MOP)

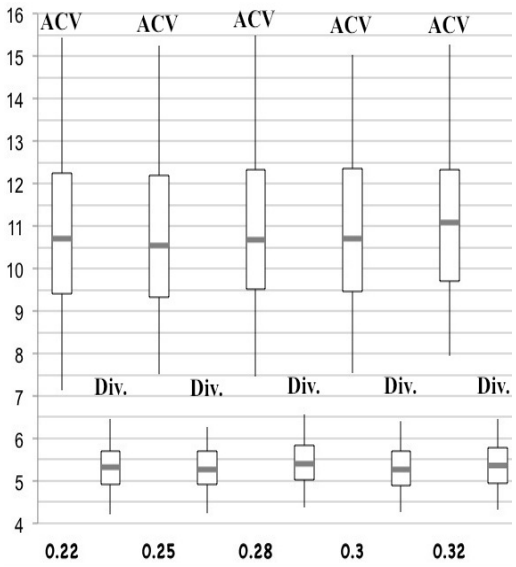

Fig. 6.41: Performance of BSselfEdges with different parameter settings γ, (4D-MOP)

ons with 2-point-edges recombination. The 4D-performance results of BSEdges, LinEdges, QuadEdges and CGainEdges with the settings $(\alpha, \omega) = (1.1, 1.5)$ are presented in Figure 6.40. Once again, QuadEdges reveal the highest metric values and CGainEdges the lowest ones. BSEdges and LinEdges achieve comparable performance results. BSselfEdges with different learning rates reveal similar performance results for all learning rate settings (Figure 6.41). In general, the metric values are of a similar level than QuadEdges. CGainEdges with different parameter settings achieves the lowest metric values of all configurations (Figure 6.42). The highest metric values are achieved with the settings $(\alpha, \omega) = (1.4, 1.8)$ compared to the other settings, but the performance differences are quite small. The increase of the parameter μ of RanEdges results in an increase of the diversity values, the ACV values tend to oscillate with an increase of μ (Figure 6.43). The performance results of RanEdges with the settings $N(3, 1.25)$ are similar to those of BSEdges and LinEdges.

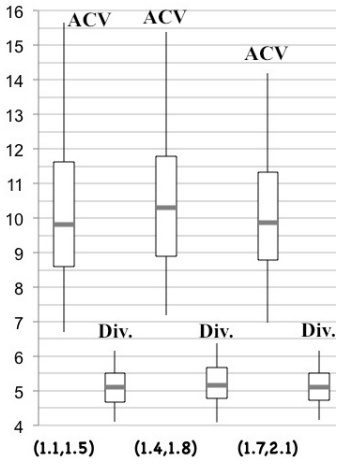

Fig. 6.42: Results of CGainEdges with different parameter settings (α, ω), (4D-MOP)

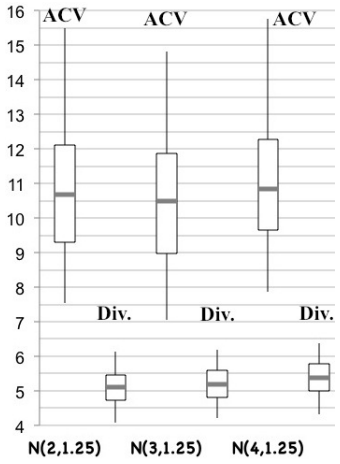

Fig. 6.43: Results of RanEdges with different parameter settings (μ, σ), (4D–MOP)

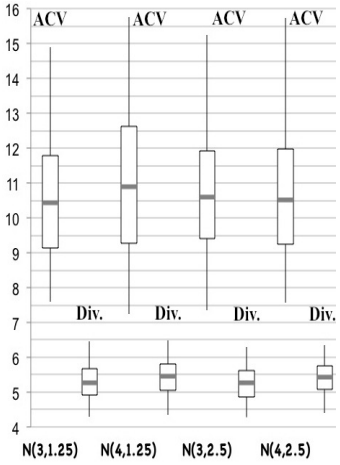

Fig. 6.44: Results of AAWeiEdges with different parameter settings (μ, σ), (4D–MOP)

The performance results of AAWeiEdges with different parameter settings reveal a tendency towards higher ones than those of RanEdges (Figure 6.44). An increase of μ results in an increase of the diversity values independent of the parameter σ, whereas no regularity is observable regarding the convergence behavior by an increase of μ for $\sigma = 1.25$ and $\sigma = 2.5$. AAWeiEdges with the settings $N(3, 1.25)$ achieves similar performance results like BSEdges and LinEdges. From a global point of view, QuadEdges achieves the highest metric values of all configurations followed by BSselfEdges. The lowest metric values are achieved by CGainEdges. BSEdges and LinEdges provide very small ACV values with respectably high diversity values at the same time. The performance values of the configurations BSEdges and LinEdges are lower than those of the configurations BSLin and LinLin.
The performance values of all configurations with 2-point-edges recombination tend to be lower than those of LiDeRP.

6.1.5 Discussion

The results of the test series presented above are summarized under the aspect of variation operator influence on the performance. The results are further discussed in a qualitative way. In general, the experiments reveal that mutation operators mainly influence convergence and diversity performance: QuadDeMut and BSself achieve the highest metric values (lowest convergence and

highest diversity) in the case of the 3D- and 4D-MOP independent of the recombination operator. Furthermore, Thierens' constant gain mutation achieves the lowest metric values (highest convergence and lowest diversity) compared to the other configurations. The configurations with BS_{adapt} result in a good convergence-diversity balance. The observations are summarized in the results matrices in the case of the 3D- and 4D-MOP. The results matrices compri-

3D-MOP	Recombinations			
Mutations	**LiDeRP**	**ExpoDeRP**	**2-point-edges**	**Random**
BSadapt	[4.23, 4.94] [5.43, 5.93]	[4.13, 4.77] [5.39, 5.88]	[3.90, 4.55] [5.21, 5.80]	[4.15, 4.83] [5.31, 5.80]
LiDeMut	[4.26, 4.95] [5.41, 5.91]	[4.22, 4.82] [5.36, 5.84]	[4.06, 4.72] [5.20, 5.78]	[4.34, 5.09] [5.37, 5.92]
QuadDeMut	[4.42, 5.06] [5.49, 5.99]	[4.46, 5.14] [5.66, 6.20]	[4.26, 4.86] [5.38, 5.90]	[4.39, 5.07] [5.53, 6.07]
Thierens (1.4,1.8)	[4.11, 4.71] [5.37, 5.88]	[3.99, 4.61] [5.23, 5.74]	[3.92, 4.67] [5.10, 5.66]	[3.87, 4.53] [5.21, 5.72]
BSself 0.22	[4.32, 4.96] [5.49, 5.96]	[4.33, 4.96] [5.44, 5.98]	[4.30, 4.96] [5.43, 5.96]	[4.36, 5.03] [4.96, 5.49]
Random (4,1.25)	[4.39, 5.02] [5.52, 6.01]	[4.22, 4.82] [5.46, 5.95]	[4.18, 4.86] [5.31, 5.92]	[4.02, 4.61] [5.46, 6.00]
AAWeighted (4,1.25)	[4.34, 4.87] [5.51, 5.96]	[4.38, 5.00] [5.53, 6.02]	[4.32, 4.99] [5.45, 5.97]	[4.48, 5.21] [5.53, 6.11]

Table 6.2: Results matrix representing the performance of VONSEA with different variation operators in the case of 3D-MOP. ACV confidence interval is the upper one in each cell, diversity confidence interval is the lower one.

4D-MOP	Recombinations			
Mutations	**LiDeRP**	**ExpoDeRP**	**2-point-edges**	**Random**
BSadapt	[9.78, 11.64] [5.21, 5.55]	[10.04, 12.02] [5.05, 5.53]	[9.30, 11.69] [5.07, 5.62]	[9.43, 11.17] [5.01, 5.51]
LiDeMut	[10.09, 12.16] [4.62, 5.16]	[10.17, 12.11] [5.18, 5.77]	[9.38, 11.54] [5.00, 5.50]	[9.59, 11.68] [5.04, 5.52]
QuadDeMut	[10.40, 12.23] [5.28, 5.78]	[10.05, 11.81] [5.20, 5.74]	[10.23, 12.18] [5.14, 5.65]	[10.32, 12.28] [5.19, 5.77]
Thierens (1.4,1.8)	[9.62, 11.43] [5.04, 5.58]	[9.42, 11.63] [5.01, 5.58]	[9.21, 11.70] [4.92, 5.51]	[9.23, 11.30] [4.96, 5.51]
BSself 0.22	[10.42, 12.39] [5.15, 5.65]	[9.84, 11.64] [5.20, 5.66]	[9.73, 11.94] [5.04, 5.60]	[10.24, 12.34] [5.19, 5.71]
Random (4,1.25)	[10.47, 12.32] [5.15, 5.62]	[10.14, 12.26] [5.22, 5.74]	[10.15, 12.05] [5.13, 5.63]	[10.11, 12.20] [5.14, 5.65]
AAWeighted (4,1.25)	[10.59, 12.46] [5.30, 5.81]	[10.35, 12.40] [5.17, 5.64]	[9.95, 12.10] [5.19, 5.69]	[9.92, 11.91] [5.11, 5.68]

Table 6.3: Results matrix representing the performance of VONSEA with different variation operators in the case of 4D-MOP. ACV confidence interval is the upper one in each cell, diversity confidence interval is the lower one.

se the confidence intervals for the mean of the VONSEA performance with various variation operators: The upper intervals in each cell represent the confidence interval of the convergence metric ACV, the lower intervals are the confidence intervals of the diversity values. The confidence intervals are determined according to the description in Section 5.3. The configurations with the default mutation parameters are used for comparison. The results of Thie-

rens' mutation scheme with the settings $\alpha = 1.4$ and $\omega = 1.8$ are presented as these settings have proven to be the best choice in many cases. The optimal parameters for BSself, Random mutation and AAWeighted depend on the configuration composition and the dimension of the MOP. The highest performance values for diversity and convergence are highlighted in red, the lowest performance values are highlighted in green. Performance results with an equally good convergence-diversity balance are highlighted in blue.
Table 6.2 presents the confidence intervals of the different configurations in the case of the 3D-MOP. The observations of the presented experimental results are reflected by the confidence intervals: The configurations with Thierens' mutation scheme are highlighted in green since they revel the lowest performance values for ACV and diversity compared to the other configurations. BSEdges is highlighted in lighter green since the confidence limits for the convergence values are relatively low compared to those of CGainEdges, but the diversity confidence limits are significantly higher. BSselfRan is also highlighted in lighter green. In this case, the diversity confidence limits tend to lower values than those of CGainRan, but the convergence confidence limits are significantly higher. BSLin, BSExpo and RanRan achieve the best convergence-diversity balance. Otherwise, QuadDeMut reveals the highest metric values in combination with LiDeRP and ExpoDeRP compared to the other configurations. AAWeighted with default settings in combination with 2-point-edges and Random recombination achieves the highest metric values compared to configurations with the other configurations.
Table 6.3 presents the confidence intervals of the VONSEA performance with various variation operators in the case of the 4D-MOP. The configurations with Thierens' mutation scheme reveal the lowest performance values compared to the other configurations, whereas QuadDeMut achieves the highest metric values. BSRan provides comparable low diversity values with a significant smaller convergence confidence interval. The configurations with AAWeighted also provide some very high performance metric values. In the case of ExpoDeRP, QuadDeMut provides significantly lower convergence values than AAWeighted but tends to higher diversity values, whereas these results are reversed in the case of 2-point-edges. In the case of Random recombination, QuadDeMut provides higher diversity values than BSself but a smaller convergence confidence interval. The configurations with BS_{adapt} reveal a good convergence-diversity balance in general.
Since the aim of these experiments is the identification of a variation operator combination that provides a constantly good convergence-diversity balance independent of the MOP, the comparison of the 3D- and 4D-MOP result

matrices reveals an accordance in the case of the configurations BSLin and BSExpo, both revealing an equally good convergence-diversity balance. As a consequence, BSLin and BSExpo have a high potential as advisable variation operator combination for similar MOPs. In general, the experiments have shown that the mutation operator has a stringent influence on the VONSEA performance and this influence is almost independent of the recombination scheme and the MOP. As assumed in the conclusions of the MOML analysis, the dynamic deterministic mutation operators comply best with the idea of supporting the search process of a MOEA in view of a specific global-local search balance. Mutation operators with varying - in the sense of oscillating - mutation rates over the performed generations like Random, AAWeighted and BSself achieve unpredictable performance results.

6.2 Selection Strategies

The following experiments investigate the search performance of the selection strategies: aggregate selection, ACV-based selection and ACV-random selection. The selection performance is analyzed regarding high solution diversity and good selection pressure[1] of the entire population at the same time. Special attention addresses the quality of the non-dominated solutions relative to the convergence behavior of the entire population. Therefore, the indicator ACV_{rel} (eq. (5.12)) is determined for each generation. The investigations comprise a fine-tuning of the selection parameters of the selection strategies: The selection probability for selecting the currently optimal individuals into the succeeding generation. The gain of the experimental results is an insight into the guidance of the search process by the selection strategies and recommendations for the parameter settings for a potential improvement of the search guidance.
The other components of the configurations are BS_{adapt} mutation, LiDeRP as recombination, a population size of 100 and three parents for recombination. BS_{adap} and LiDeRP have been identified as a good variation operator combination with regard to a constantly good convergence-diversity balance and are therefore used for the following investigations. The indicator ACV_{scaled} (eq. (5.11)) is used as convergence metric for a more differentiated consideration of the convergence behavior, since all objective values have a scaling-

[1] The term 'selection pressure' is used to characterize the strong (high selection pressure) or respectively weaker (smaller selection pressure) emphasis of selection on the best individuals. [7]

independent influence on the indicator values. Special attention is paid to the non-dominated solutions in each generation. Boxplots are presented to compare the relative quality of the non-dominated solutions identified by the different configurations.

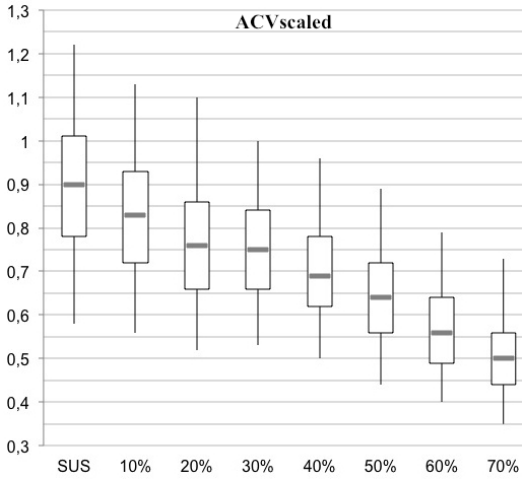

Fig. 6.45: Performance of aggregate selection with p_0 variation. (3D-MOP)

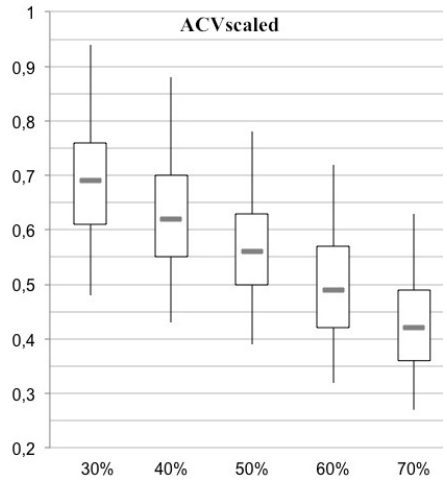

Fig. 6.46: Performance of ACV-based selection with p_0 variation. (3D-MOP)

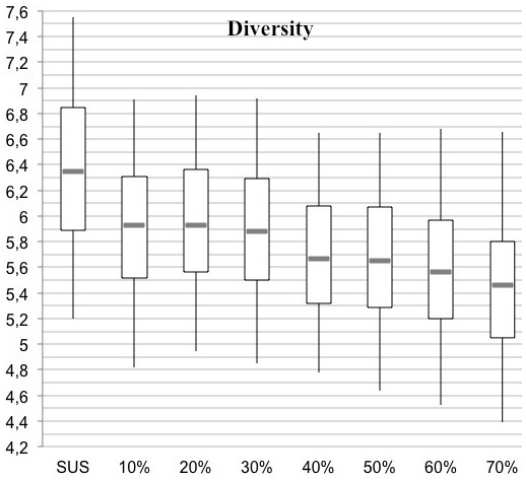

Fig. 6.47: Diversity of aggregate selection with a p_0 variation. (3D-MOP)

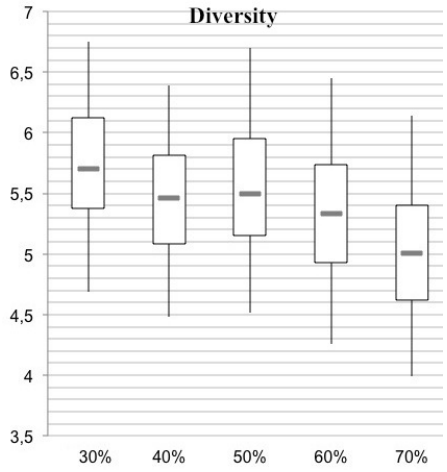

Fig. 6.48: Diversity of ACV-based selection with p_0 variation. (3D-MOP)

6.2.1 Variation of the Selection Probability by Fixed Tournament Size

The first series of tests investigate the performance of the three selection strategies with a variation of the selection probability parameter p_0 for selecting

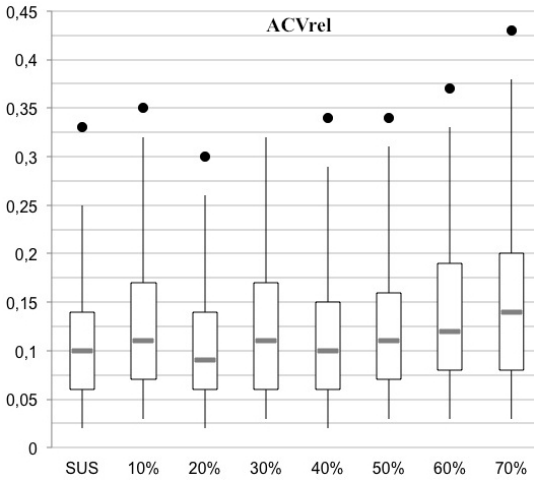

Fig. 6.49: ACV_{rel} of aggregate selection with a p_0 variation. (3D-MOP)

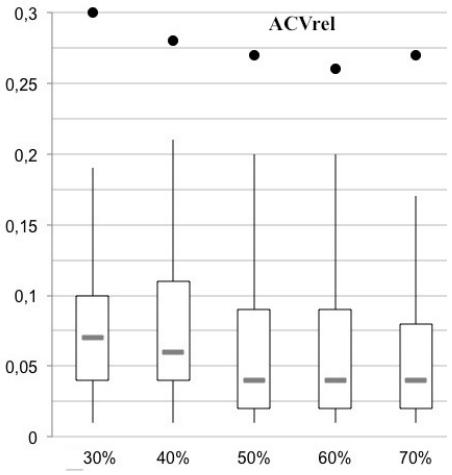

Fig. 6.50: ACV_{rel} of ACV-based selection with p_0 variation. (3D-MOP)

currently optimal individuals into the succeeding generation. Different probabilities are tested with the fixed tournament size parameter $t.s. = 10$, firstly in the case the 3D-MOP. Figure 6.45 represents the effect on the convergence performance of the aggregate selection with varying probability values p_0. The increase of p_0 results in a stringent decrease of the ACV_{scaled} values at different intensities. The increase of p_0 from 0% to 20% results in the strongest and most continuous decrease of the ACV_{scaled} values. For $p_0 = 30\%$, the ACV_{scaled} results are nearly comparable to $p_0 = 20\%$ and a further increase of p_0 results in an attenuated decrease of the ACV_{scaled} values. The configuration with the probability value $p_0 = 0\%$ - denoted as SUS in the figures - achieves a remarkably high spread of the ACV_{scaled} values indicating an unstable selection process. In general, the decrease of p_0 also results in a continuous decrease of the ACV_{scaled} value spread, which is observable by the decrease of the interquartile range.

Figure 6.47 represents the diversity achieved by the configurations with aggregation selection and varying p_0 values. The configuration with SUS selection achieves highest diversity. The further increase of p_0 results in an oscillating decrease of diversity. Furthermore, the SUS configuration also achieves the highest spread of diversity values at all. The diversity is significantly decreased by an increase of p_0 to 10%. The diversity results of $p_0 = 30\%$ are comparable to those of $p_0 = 10\%$, the same is valid for the configurations with $p_0 = 40\%$ and $p_0 = 50\%$. The lowest diversity results are achieved with $p_0 = 70\%$. In general, the comparison of Figure 6.45 to 6.47 reveals that the convergence improvement is once again at the cost of diversity. As the convergence improvement of $p_0 = 50\%$ to $p_0 = 60\%$ is on average higher than the average decrease of

the diversity values for the same p_0 values, the optimal choice is regarded as $p_0 = 60\%$.

The relative quality results of the non-dominated solutions achieved by the configurations with aggregate selection and varying probability values is presented in Figure 6.49, which depicts the ACV_{rel} values of the different configurations. The lowest ACV_{rel} values on average are achieved by the configurations with SUS and $p_0 = 20\%$. This indicates that the ACV_{rel} values of the non-dominated solutions are on average relatively low compared to the ACV-values of the entire population. The configurations with the probability values of $p_0 = 0\%$ to $p_0 = 30\%$ reveal oscillating ACV_{rel} values. The further increase from $p_0 = 40\%$ results in a continuous increase of the ACV_{rel} values. This indicates increasing ACV values of the non-dominated solutions compared to the ACV values of the entire population. Furthermore, the spread of the ACV_{rel} values is also increasing, observable by the increasing inter-quartile ranges. The highest ACV_{rel} values are achieved by the configuration with $p_0 = 70\%$. Nearly all configurations reveal upwards outliers for the ACV_{rel} values, which indicates that there are significantly higher values.

These results of aggregate selection strategy reveal that the configurations with sole front-based SUS or a high probability for selecting individuals into the succeeding generation by front-based SUS results in unsteady performance with regard to convergence and diversity. The outcomes of the relative non-dominated solution quality reveal that an increase of the first-front-based selection of the tournament set does not guarantee the selection of the highest quality solutions, since no continuous improvement of the ACV_{rel} values is observable by an increase of p_0.

Figure 6.46, 6.48 and 6.50 present the performance results of the configurations with ACV-based selection and a variation of the selection probability. The ACV_{scaled} values presented in Figure 6.46 reveal once again a stringent decrease of the ACV_{scaled} values with an increase of the probability from $p_0 = 30\%$ to $p_0 = 70\%$. The comparison of convergence performance of the aggregate selection to the ACV-based selection reveals that the performance results of the configuration with aggregate selection and $p_0 = 40\%$ are equal to those of ACV-based selection with $p_0 = 30\%$. Moreover, the performance results of aggregate selection with $p_0 = 50\%$ and $p_0 = 60\%$ are comparable to those of ACV-based selection with $p_0 = 40\%$ and $p_0 = 50\%$, respectively. Otherwise, the ACV-based selection convergence performance of $p_0 = 70\%$ tends to be lower than the aggregate selection convergence performance of $p_0 = 60\%$. These results indicate that the ACV-based selection criterion is more reliable in selecting high quality solutions than the rank-based selection criterion.

The diversity values of the ACV-based configurations are presented in Figure 6.48. Once again, the improved convergence performance is achieved at the cost of diversity. The increase of the selection probability results in a steady decrease of the diversity values except from the configuration with $p_0 = 40\%$, which represents an outlier. The comparison of the diversity performance of the ACV-based selection with $p_0 = 30\%, 50\%, 60\%$ to the one of aggregate selection with $p_0 = 40\%, 60\%, 70\%$ exposes that the diversity values are highly related. The relative quality of the non-dominated solutions detected by the configurations with ACV-based selection is depicted in Figure 6.50. The highest ACV_{rel} values are achieved by the configuration with $p_0 = 40\%$. In general, an increase of the selection probability results in a decrease of the ACV_{rel} values. Furthermore, every configuration provides upwards outliers revealing significantly higher values. The ACV_{rel} performance of the configurations with $p_0 = 50\%$ and 60% are equal. The decrease of the ACV_{rel} values in the light of the convergence improvement of the entire population in the case of p_0 increase indicates that the ACV-based selection guarantees the selection of the highly qualified solutions to a corresponding level. The comparison the the ACV_{rel} performance of the ACV-based selection to those of the aggregate selection exposes that the ACV_{rel} values of the ACV-based selection are generally lower than the corresponding values of the aggregate selection, though the convergence results of the entire population are comparable for different p_0 settings. Since the aggregate selection performance with $p_0 = 60\%$ is comparable to the performance results of the ACV-based selection with $p_0 = 50\%$ with regard to convergence and diversity and the ACV_{rel} values of the ACV-based selection are significantly improved compared to aggregate selection, $p_0 = 50\%$ is regarded as the optimal parameter setting for ACV-based selection and the advisable selection configuration in general.

Figure 6.51, 6.53 and 6.55 present the performance results of the configurations with ACV-random selection and a variation of the selection probability. The convergence performance depicted in Figure 6.51 exposes a stringent decrease of the ACV_{scaled} values by an increase of the selection probability. Once again, the probability decrease for selecting individuals via front-based SUS results in a decrease of the ACV_{scaled} value spread, observable by the decrease of the inter-quartile ranges. The diversity performance of the configurations with ACV-random selection depicted in Figure 6.53 reveals a similar decrease of the diversity values by an increase of p_0. The comparison of the convergence performance of ACV-random selection to the corresponding aggregate (Figure 6.45) and ACV-based selection performance (Figure 6.46) exposes that the convergence values of the ACV-random selection performance are general-

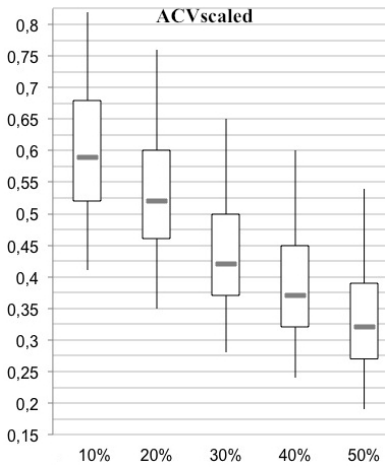

Fig. 6.51: Performance of ACV-random selection with p_0 variation. (3D-MOP)

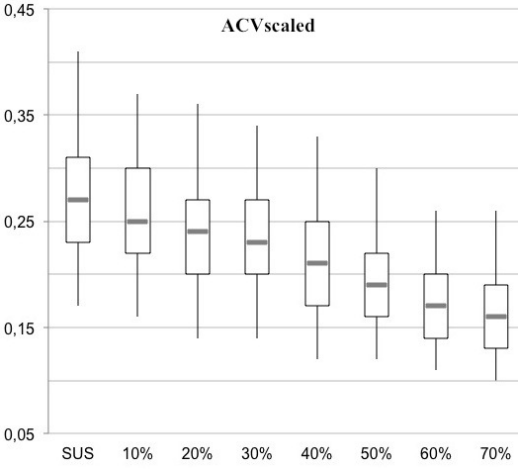

Fig. 6.52: Performance of aggregate selection with p_0 variation. (4D-MOP)

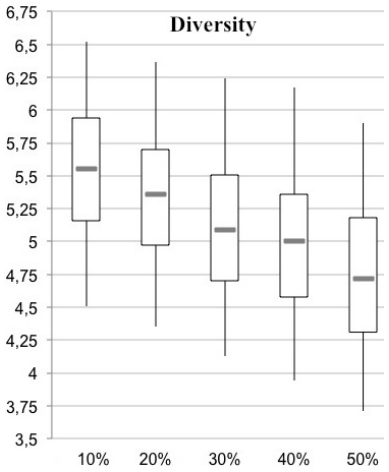

Fig. 6.53: Diversity of ACV-random selection with p_0 variation. (3D-MOP)

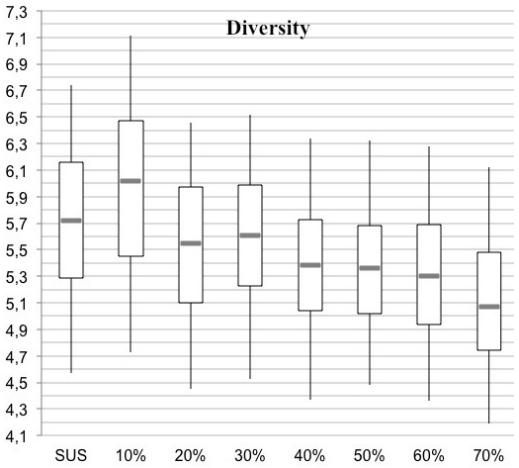

Fig. 6.54: Diversity of aggregate selection with p_0 variation. (4D-MOP)

ly significantly lower than those of the ACV-based and aggregate selection performance for high p_0 settings: More precise, the ACV-random convergence performance with $p_0 = 10\%$ is comparable to aggregate selection convergence performance with $p_0 = 50\%$ and to ACV-based selection performance with $p_0 = 40\%$. Furthermore, ACV-random selection with $p_0 = 20\%$ achieves a convergence performance between the performance results of aggregate selection with $p_0 = 60\%$ (advisable setting) and 70%. The ACV-random results with $p_0 = 30\%$ are comparable to aggregate selection with $p_0 = 70\%$. The comparison results of the diversity performance are similar: The diversity performance of ACV-random selection with $p_0 = 10\%$ is comparable to the results of aggregate selection with $p_0 = 60\%$ (Figure 6.47) and to ACV-based selection with the advisable setting $p_0 = 50\%$ (Figure 6.48). Furthermore, the ACV-random

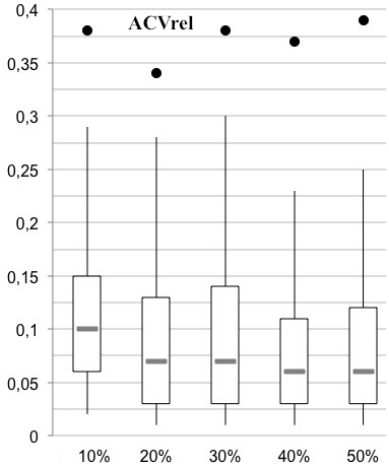

Fig. 6.55: ACV_{rel} of ACV-random selection with p_0 variation. (3D-MOP)

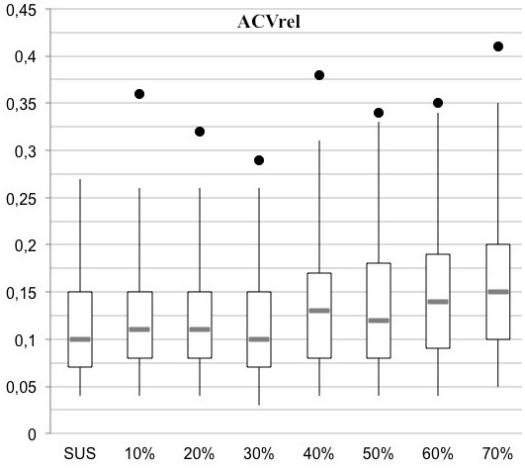

Fig. 6.56: ACV_{rel} of aggregate selection with p_0 variation. (4D-MOP)

performance with $p_0 = 20\%$ tends to be lower than the performance of aggregate selection and the ACV-random performance of $p_0 = 30\%$, which is slightly higher than the performance of ACV-based selection with $p_0 = 70\%$. Figure 6.55 presents the performance results of the relative non-dominated solution quality of the configurations with ACV-random selection. The increase of the selection probability results tends to an oscillating decrease of the ACV_{rel} values. Once again, all configurations provide outliers revealing significantly higher ACV_{rel} values. The range of ACV_{rel} values for the configuration with $p_0 = 30\%$ is the highest of all configurations and the ACV_{rel} values for $p_0 = 40\%$ tend to be the lowest for all configurations. The comparison of the ACV_{rel} values of ACV-random selection to the ACV_{rel} values of aggregate (Figure 6.49) and ACV-based selection (Figure 6.50) reveals that the level of ACV_{rel} values is generally lower than those of aggregate selection, but tends to be higher than those of ACV-based selection in general. As the highest diversity values are achieved for the ACV-random configurations with $p_0 = 10\%$ with the lowest ACV_{rel} values for the same configuration, this configuration is regarded as the advisable setting for the 3D-MOP. In general, the results of the ACV-random selection performance reveal the worst overall performance. This empirically verifies that the fitness-proportionate selection is superior to a random individual selection.

The presentation of the following experimental results refers to the performance of the selection strategies with a variation of the selection probability in the case of the 4D-MOP. The results are additionally discussed in the light of the

corresponding performance results of the 3D-MOP.
Figure 6.52, 6.54 and 6.56 present the 4D-performance results of configurations with aggregate selection and a variation of the selection probability. The convergence performance of these configurations is depicted in Figure 6.52. Similar to the 3D-performance results of the configurations with aggregate selection (Figure 6.45), an increase of the selection probability results in a continuous decrease of the ACV_{scaled} values. Equally to the 3D-performance results, the ACV_{scaled} results of the configurations with $p_0 = 20\%$ and 30% are comparable. The spread of the ACV_{scaled} values of the different configurations slightly decreases with an increase of p_0. This level of decrease is lower than in the case of the corresponding 3D-performance results as a consequence of the higher percentage of the non-dominated solutions and the lower front diversity, as documented in the landscape analysis chapter 3.7. Figure 6.54 depicts the diversity results of the different configurations with aggregate selection. Similar to the corresponding 3D-diversity results (Figure 6.47), an increase of the selection probability results in an oscillating decrease of the diversity values, albeit to a lesser extent. In contrast to the 3D-results, the tendency towards highest diversity values are achieved for the configuration with $p_0 = 10\%$ instead of the SUS configurations. The diversity results of the configurations with $p_0 = 40\%$ and 50% are comparable. The relative quality results of the non-dominated solutions achieved by configurations with aggregate selection and varying probability values are presented in Figure 6.56. Similar to the corresponding 3D-results (Figure 6.49), there is a continuous increase of the ACV_{rel} values observable by an increase of p_0 starting at $p_0 = 40\%$ indicating that the ACV values of the non-dominated solutions increase more sharply relative to the ACV values of the entire population. Nearly all configurations provide outliers, who are closest to the maximal value. The tendency towards lowest ACV_{rel} values are achieved for configurations with p_0 lower than 40%. As the diversity values of the configurations with $p_0 = 40\%$ and 50% are comparable, a slight decrease of the diversity values achieved by the configuration with $p_0 = 60\%$ is observable in comparison to the configurations with $p_0 = 40\%, 50\%$. The ACV_{scaled} decrease by an increase of $p_0 = 50\%$ to 60% is significant. Therefore, the setting of $p_0 = 60\%$ is advisable.
Figure 6.57, 6.59 and 6.61 present the 4D-performance results of the configurations with ACV-based selection and a variation of the selection probability. The convergence performance depicted in Figure 6.57 exposes a stringent decrease of the ACV_{scaled} values by an increase of the selection probability and therefore reveals similar performance results as in the case of the 3D-performance results (Figure 6.46). Figure 6.59 presents the diversity results: Configurati-

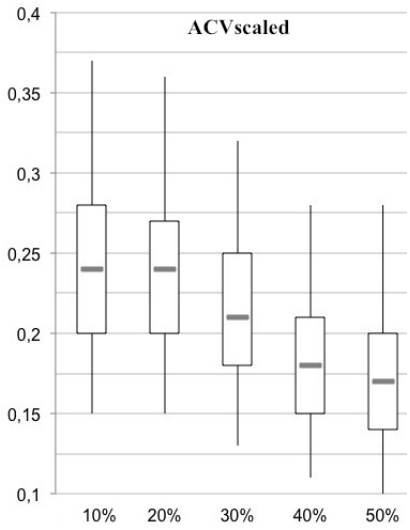

Fig. 6.57: Performance of ACV-based selection with p_0 variation. (4D-MOP)

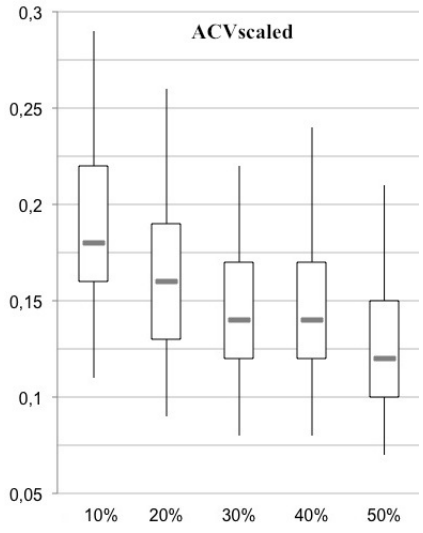

Fig. 6.58: Performance of ACV-random selection with p_0 variation. (4D-MOP)

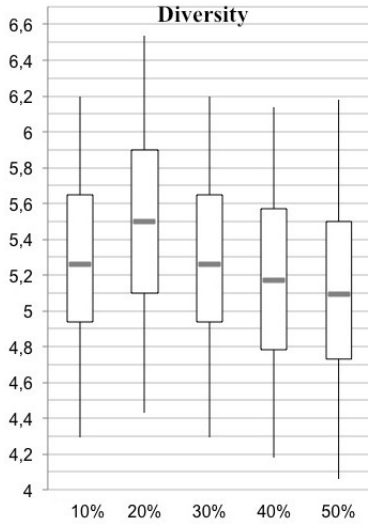

Fig. 6.59: Diversity of ACV-based selection with p_0 variation. (4D-MOP)

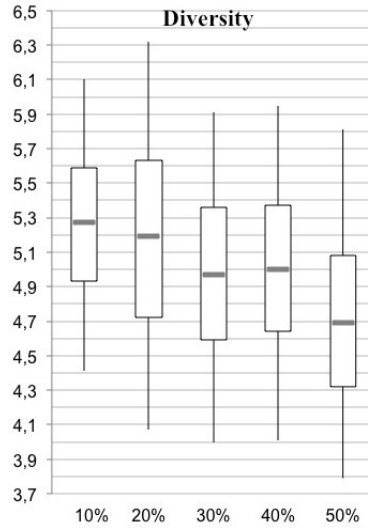

Fig. 6.60: Diversity of ACV-random selection with p_0 variation. (4D-MOP)

ons with $p_0 = 10\%$ and 30% are comparable in the performance results. The configuration with $p_0 = 20\%$ achieves the highest diversity values. Configurations starting by $p_0 = 30\%$ reveal a continuous decrease of the diversity values. The comparison of these results to the performance results of aggregate selection reveals lower convergence performance on average for the ACV-based selection (Figure 6.52). The comparison of the diversity values reveals that the performance results with $p_0 = 10\%$ and 30% are similar to those of aggregate selection with $p_0 = 40\%$ and 50%. The relative non-dominated solution quality is presented in Figure 6.61. The ACV_{rel} values continuously decrease

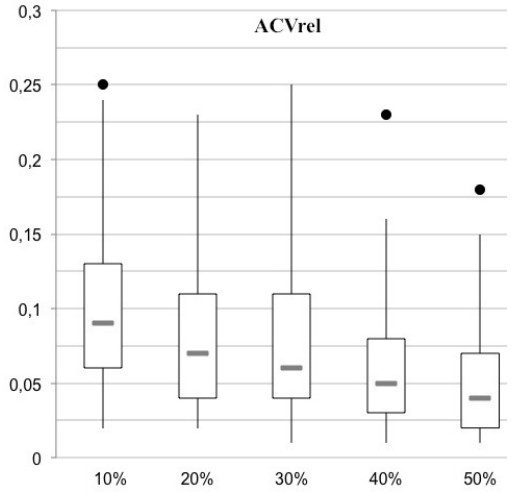

Fig. 6.61: ACV_{rel} of ACV-based selection with p_0 variation. (4D-MOP)

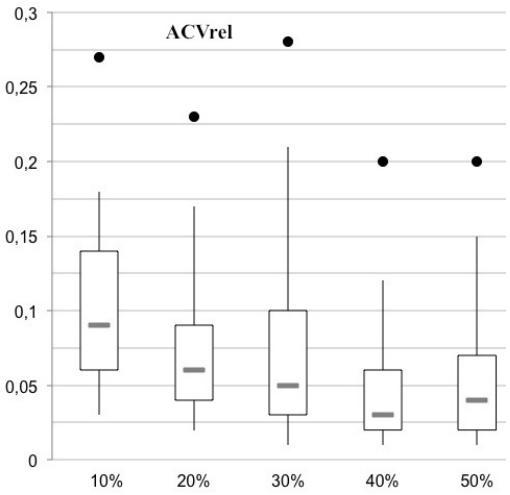

Fig. 6.62: ACV_{rel} of ACV-random selection with p_0 variation. (4D-MOP)

by an increase of the probability values. The ACV_{rel} values of the configurations $p_0 = 20\%$ and 30% are nearly comparable. The comparison of these performance results to the ACV_{rel} values of the aggregate selection (Figure 6.56) reveals that the values of configurations with ACV-based selection are in general significantly lower. The convergence results of ACV-based selection with $p_0 = 50\%$ tend to be lower than the convergence performance of aggregate selection with $p_0 = 60\%$, but the diversity values of these configurations with ACV-based selection are remarkably lower than those of configurations with aggregate selection. Furthermore, the ACV_{rel} values of ACV-based selection are significantly lower than those of aggregate selection for the latter mentioned probability values. The clear performance advantage of ACV-based selection over aggregate selection - as described in the case of the 3D-MOP - is not given by these observations. Therefore, the question concerning an optimal selection strategy for the 4D-MOP is a question concerning a trade-off between higher diversity and higher relative non-dominated solution quality.

The 4D-performance results of the configurations with ACV-random selection and various probability values are presented in the Figure 6.58, 6.60 and 6.62. The convergence performance reveals once more a continuous decrease of the ACV_{scaled} values with an increase of the probability values (Figure 6.58). The convergence performance results of the configurations with $p_0 = 30\%$ and 40% are similar. The diversity values also reveal a nearly continuous decrease of the diversity values with an increase of p_0. The configurations with $p_0 = 20\%$ tend to have the highest diversity values. The performance results of the configurations with $p_0 = 30\%$ and 40% are comparable, though the configurations with $p_0 = 40\%$ provide slightly increased diversity values. Figure 6.62 presents

the relative non-dominated solution quality of the configurations with ACV-random selection. In general, the performance results of ACV_{rel} decrease in an oscillating manner. The configurations with $p_0 = 10\%$ and 30% provide a wide spread of ACV_{rel} values, which is observable by higher inter-quartile ranges as well as outliers. All configurations provide outliers, partly in a remarkable distance from the maximal value. The comparison of these performance results to the 4D-results of aggregate and ACV-based selection reveals that ACV-random selection provides the worst performance results: ACV-random configurations achieve in general a tendency towards lower convergence performance compared to the configurations with aggregate selection (Figure 6.52), the ACV-random configuration with $p_0 = 10\%$ is comparable in convergence performance to the aggregate configurations with $p_0 = 50\%$. A further increase of p_0 starting from 10% results in lower ACV_{scaled} values. Compared to the ACV-based configurations (Figure 6.57), the convergence performance of the ACV-random configurations tends to be higher than those of the ACV-based configurations. The configurations of the selection strategies with $p_0 = 50\%$ are comparable. Therefore, the convergence performance of ACV-random selection lies somewhere in-between those of aggregate and ACV-based selection. On the other side, the diversity performance of the ACV-random configurations is in general lower than the diversity of aggregate and ACV-based selection. Only, the configurations results of ACV-random selection with $p_0 = 10\%$ are of the same level than the one of ACV-based configuration with $p_0 = 50\%$. Moreover, the quality performance results of the ACV-random configurations are in general significantly lower than those of aggregate selection, but tend to be higher than those of ACV-based selection. The ACV_{rel} results of ACV-random and ACV-based selection for $p_0 = 10\%$ are comparable. Therefore, ACV-random is not able to outperform the ACV-based or aggregate selection, but the parameter $p_0 = 10\%$ in the case of ACV-random selection is regarded as advisable for the 4D-MOP.

6.2.2 Variation of the Tournament Size by Fixed Selection Probability

The following series of tests investigate the configurations with different selection strategies and a variation of the parameter ts with the aim of a further performance improvement in terms of a good convergence-diversity balance and a high relative non-dominated solution quality. The advisable selection probability settings p_0 of each selection strategy, which are considered as the

most advisable according to the experiments presented above, are applied for a further performance improvement by a variation of the parameter ts. The expected results are a closer insight into the performance changes by an increase or decrease of the default setting $ts = 10$ in the case of the 3D- and 4D-MOP. Moreover, the hypothesis with regard to the interdependence of the tournament size and selection pressure is empirically verified: the selection pressure is modifiable by varying the ts setting. More precisely, the increase of ts results in a higher selection pressure.

Figure 6.63, 6.65 and 6.67 present the 3D-performance results of the confi-

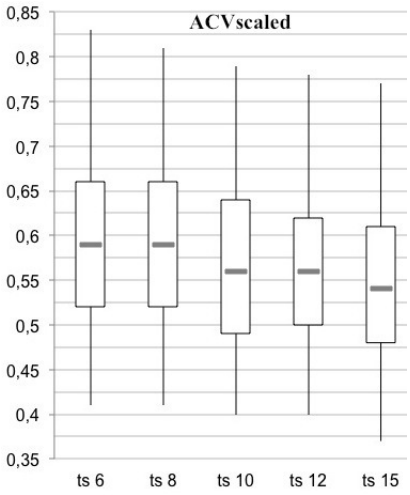

Fig. 6.63: Performance of aggregate selection with ts variation and $p_0 = 60\%$. (3D-MOP)

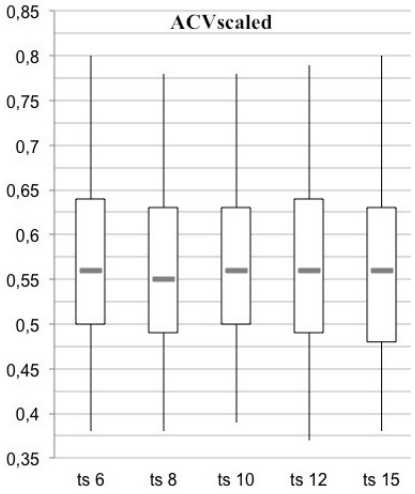

Fig. 6.64: Performance of ACV-based selection with ts variation and $p_0 = 50\%$. (3D-MOP)

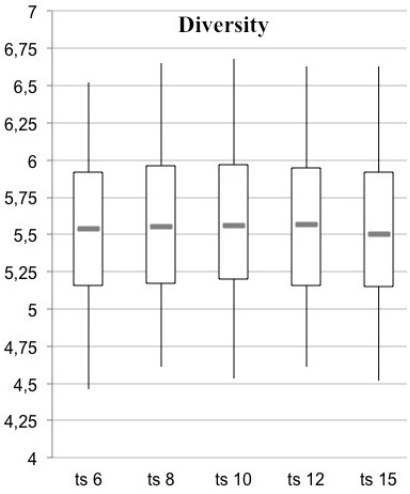

Fig. 6.65: Diversity of aggregate selection with ts variation and $p_0 = 60\%$. (3D-MOP)

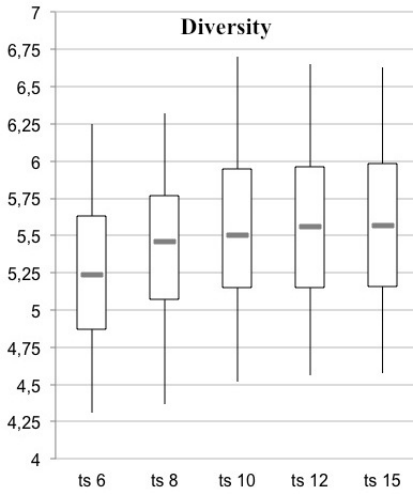

Fig. 6.66: Diversity of ACV-based selection with ts variation and $p_0 = 50\%$. (3D-MOP)

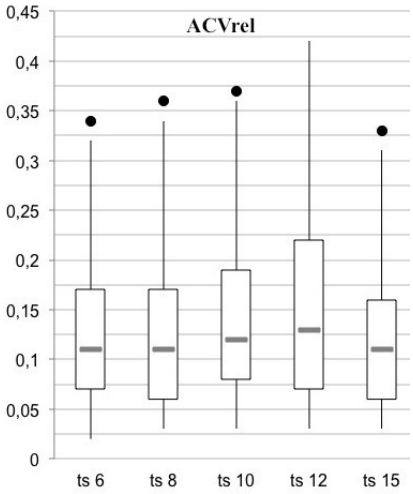

Fig. 6.67: ACV_{rel} of aggregate selection with ts variation and $p_0 = 60\%$. (3D-MOP)

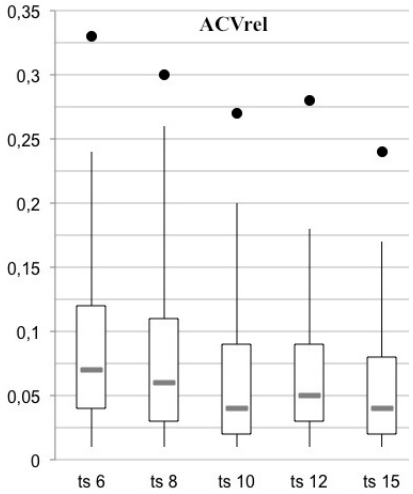

Fig. 6.68: ACV_{rel} of ACV-based selection with ts variation and $p_0 = 50\%$. (3D-MOP)

gurations with aggregate selection and a variation of ts. The increase of ts starting by 10 results in an improvement of the convergence performance by a decrease of the ACV_{scaled} values as depicted in Figure 6.63. The convergence performance of the configurations with $ts < 10$ are equal. This indicates that the higher solution number in the tournament set provides a more discerning selection of an individual from the first front. The diversity of these configurations depicted in Figure 6.65 reveals on average only very slight changes of the values by the increase of ts. The tendency towards highest diversity values are achieved for $ts = 10$. The relative non-dominated solution quality, as presented in Figure 6.67, reveals comparable ACV_{rel} values for the configurations with $ts = 6$ and 8. The highest ACV_{rel} values are achieved for $ts = 12$, the tendency towards very low values is achieved by the configurations with $ts = 15$. Nearly all configurations provide outliers, which are very close to the maximal values. In general, the question for the advisable ts value is a trade-off between a higher computational complexity in every iteration caused by the ranking of the higher number of solutions in the tournament set and the performance improvement regarding the convergence-diversity balance as well as the relative non-dominated solution quality. According to these results, $ts = 10$ is regarded as advisable setting in the case of aggregate selection.

Figures 6.64, 6.66 and 6.68 present the 3D-performance results of the configurations with ACV-based selection and different ts values. The convergence results of these configurations are generally comparable among each other, since the differences between the ACV_{scaled} values of the various ts-configurations are only slight. The lowest range of ACV_{scaled} values are achieved for the confi-

guration with $ts = 10$. The tendency towards lowest ACV_{scaled} values in general are achieved for $ts = 15$, since the higher number of solutions in the tournament set provides a more discerning selection within the solutions with the lowest ACV_{scaled} values. The diversity performance of the ACV-based configurations is presented in Figure 6.66. The increase of the ts reveals an improvement of diversity up to $ts = 12$, the diversity results of the configurations $ts = 12$ and 15 are similar. The highest improvement is achieved by the increase of $ts = 6$ to 10. The relative non-dominated solution quality is depicted in Figure 6.68 and reveals a decrease of the ACV_{rel} values by an increase of ts, only the performance of the configuration $ts = 12$ are slightly increased in this row. The nearly continuous decrease of the ACV_{rel} in the light of the convergence results of the entire population is once more a consequence of the increasing solution number in the tournament set providing a more and more discerning selection. An advisable setting for ts is - as a consequence of these results - the choice of $ts = 10$ as a trade-off between a good convergence-diversity performance, high relative non-dominated solution quality and an acceptable computational complexity in each iteration step. The computational complexity is mainly caused by the calculation of the ACV_{scaled} values of all solutions in the tournament set, which is comparably lower than the ranking of the these solutions. Furthermore, the comparison results of the configurations with aggregate and ACV-based selection reveals once again an advantage of ACV-based selection over aggregate selection in the case of $ts = 10$ for both configurations: The convergence performance of the configurations with AVC-based selection are on the same level than the convergence performance of aggregate configurations with $ts = 10$ (Figure 6.63). Moreover, the diversity performance of configurations with ACV-based selection and ts starting by 10 is comparable to the diversity performance of aggregate selection (Figure 6.65). The main advantage of ACV-based selection is the exceptionally high relative non-dominated solution quality, which is generally significantly better than for the configurations with aggregate selection. Therefore, ACV-based selection with $ts = 10$ is regarded as superior to configurations with aggregate selection.

The performance results of the configurations with ACV-random selection and a variation of ts are presented in Figure 6.69, 6.71 and 6.73. In general, the variation of ts does not reveal an observable performance alteration: The convergence performance of the configurations are nearly comparable, the tendency towards lowest ACV_{scaled} values is achieved for $ts = 8$ and tend to be the highest for $ts = 10$ as well as 12. Diversity is improved by an increase of ts up to $ts = 10$ (Figure 6.71), but decreases afterwards. Therefore, the best diversity performance is achieved for $ts = 10$. The relative non-dominated solution qua-

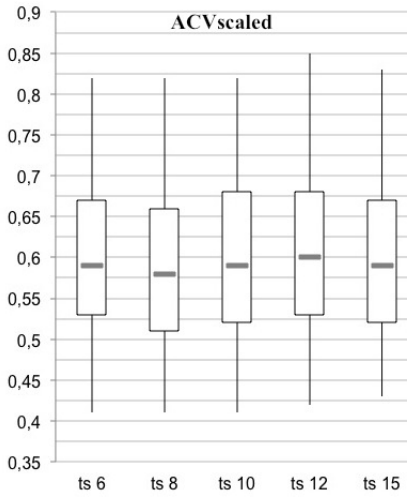

Fig. 6.69: Performance of ACV-random selection with ts variation and $p_0 = 10\%$. (3D-MOP)

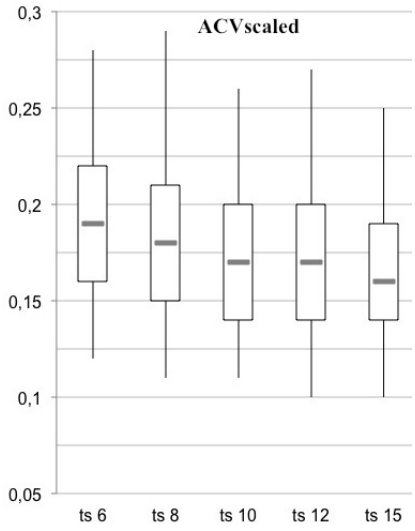

Fig. 6.70: Performance of aggregate selection with ts variation and $p_0 = 60\%$. (4D-MOP)

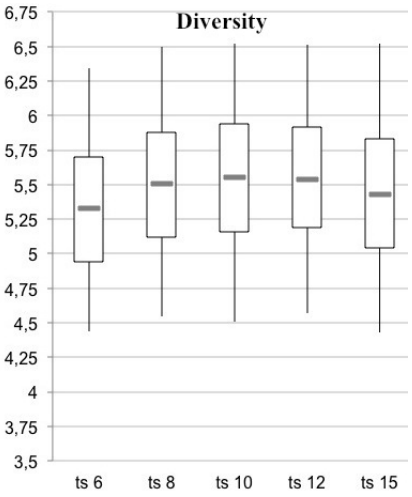

Fig. 6.71: Diversity of ACV-random selection with ts variation and $p_0 = 10\%$. (3D-MOP)

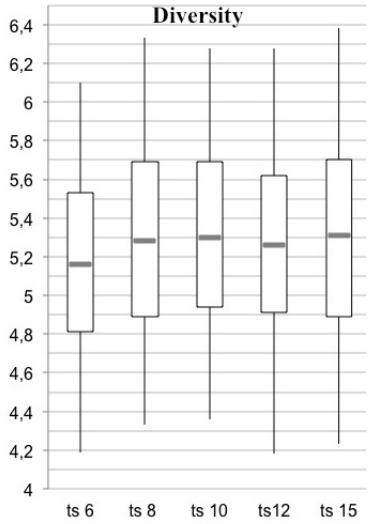

Fig. 6.72: Diversity of aggregate selection with ts variation and $p_0 = 60\%$. (4D-MOP)

lity is continuously improved with an increase of ts. The lowest range of ACV_{rel} values is achieved for $ts = 10$. All configurations provide outliers indicating in some cases remarkably higher values. These results reveal that the variation of the tournament size has no clear effect neither on the convergence nor on the diversity in the case of ACV-random selection with $p_0 = 10\%$. Furthermore, configurations with ACV-random selection do not provide the opportunity to outperformance the configurations with ACV-based or aggregate selection by a variation of ts.

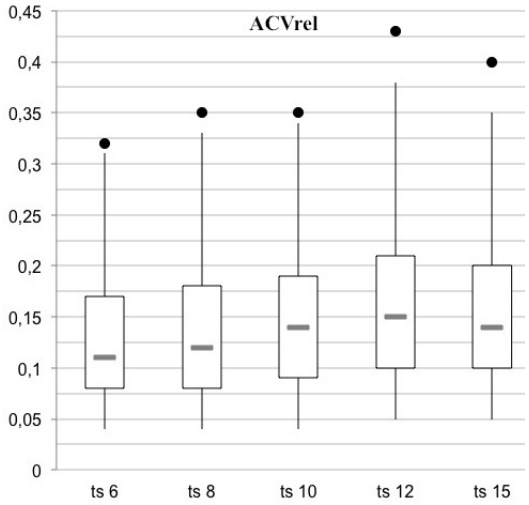

Fig. 6.73: ACV_{rel} of ACV-random selection with ts variation and $p_0 = 10\%$. (3D-MOP)

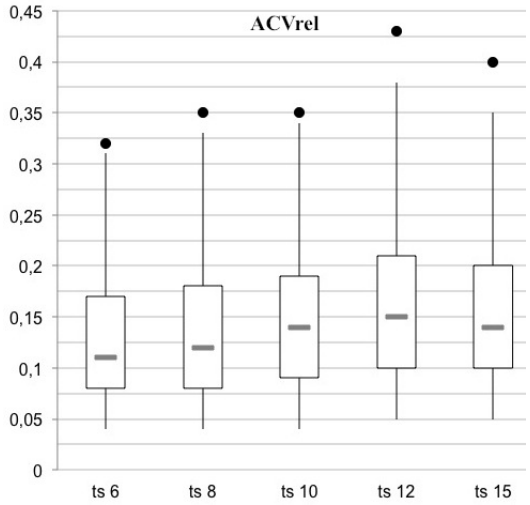

Fig. 6.74: ACV_{rel} of aggregate selection with ts variation and $p_0 = 60\%$. (4D-MOP)

Summarizing, the influence of the selection parameter variation on the performance of the different configurations depends on the specific selection strategy. In the case of the configurations with aggregate selection, higher values of ts result in an improved convergence performance as a consequence of the more discerning selection caused by the larger range of solutions in the tournament set. However, the diversity is stagnating and the relative non-dominated solution quality is not generally improved. This demonstrates the effect that the random selection of a first-front solution does not guarantee the selection of the fittest individual according to all objectives. In the case of configurations with ACV-random selection, lower values of ts result in a decrease of the diversity values as well as a worse relative non-dominated solution quality. Otherwise, higher values of ts achieve stable convergence and diversity performance with significantly increased relative non-dominated solution quality. ACV-based selection overcomes the disadvantage of aggregate selection, since the larger solution range in the tournament set provides a higher range of high quality solutions to be selected into the succeeding generation. No regular influence of ts variation is observed in the case of ACV-random selection. This is a consequence of the fact that random individual selection is not a discerning selection strategy, which is generally affected by a larger solution range in the tournament set.

The following experiments present the performance of the configurations with different selection strategies and a variation of ts in the case of the 4D-MOP.

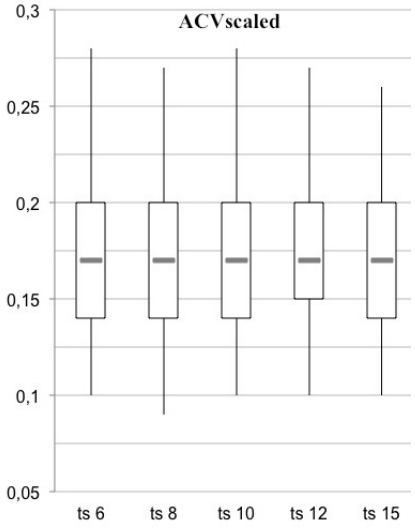

Fig. 6.75: Performance of ACV-based selection with ts variation and $p_0 = 50\%$. (4D-MOP)

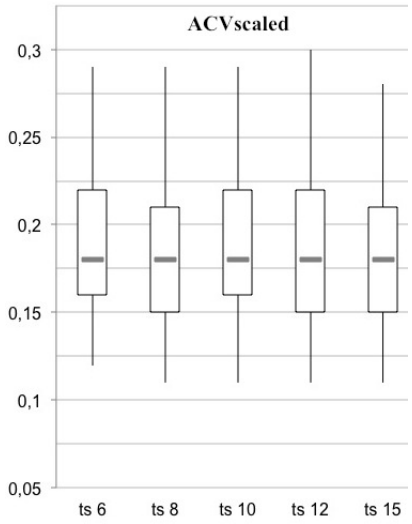

Fig. 6.76: Performance of ACV-random selection with ts variation and $p_0 = 10\%$. (4D-MOP)

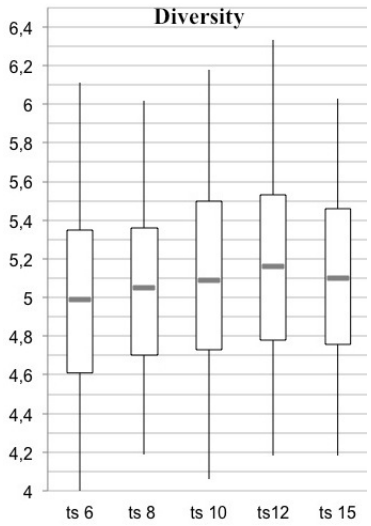

Fig. 6.77: Diversity of ACV-based selection with ts variation and $p_0 = 50\%$. (4D-MOP)

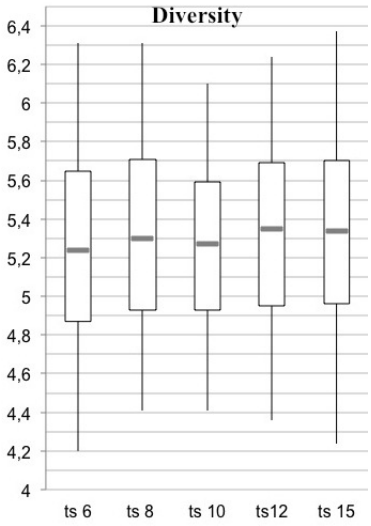

Fig. 6.78: Diversity of ACV-random selection with ts variation and $p_0 = 10\%$. (4D-MOP)

The results are additionally discussed in the light of the observations of the selection strategy-specific parameter influence in the case of the 3D-MOP. Figure 6.70, 6.72 and 6.74 present the 4D-performance results of the configurations with aggregate selection and a ts variation. An increase of ts results in continuous decrease of the ACV_{scaled} values and therefore in improved convergence behavior (Figure 6.70). These convergence results are comparable to those of the corresponding 3D-results of the configurations with aggregate selection (Figure 6.63). Furthermore, the convergence results of $ts = 10$ and 12

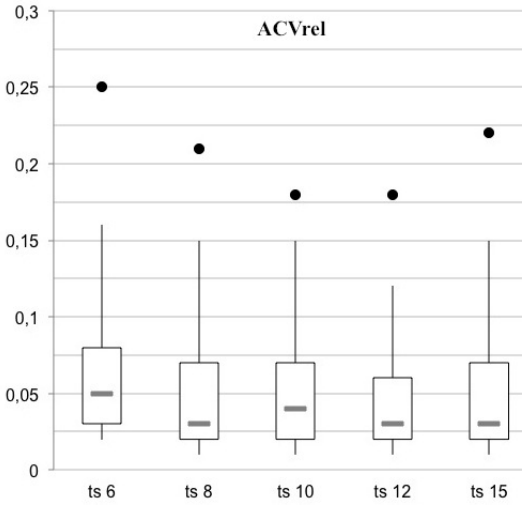

Fig. 6.79: ACV_{rel} of ACV-based selection with ts variation and $p_0 = 50\%$. (4D-MOP)

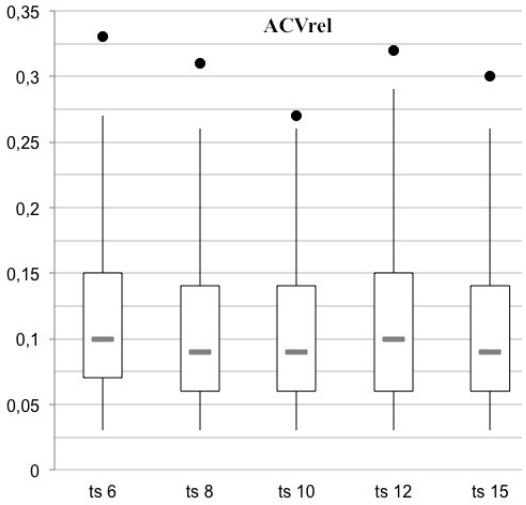

Fig. 6.80: ACV_{rel} of ACV-random selection with ts variation and $p_0 = 10\%$. (4D-MOP)

are comparable. The diversity results reveal no regularity by an increase of ts (Figure 6.72). The tendency towards highest diversity values is achieved for $ts = 8, 10$ and 15, lowest diversity is achieved for $ts = 6$. The relative non-dominated solution quality of these configurations is presented in Figure 6.74. The ACV_{rel} values are increased by an increase of ts. The tendency towards the highest ACV_{rel} values are achieved with $ts = 12$. All configurations provide outliers, which are mostly close to the maximal values. Once again, the relative non-dominated solution quality is comparable to the corresponding 3D-results (Figure 6.67). The configurations with $ts = 10$ achieve once again a good convergence-diversity balance with a tendency towards highest diversity values and the second lowest convergence values as well as a relative non-dominated solution quality on a medium level in comparison to the other configurations. Therefore, the configuration with aggregate selection $ts = 10$ is regarded as an advisable parameter setting.

The performance results of the configurations with ACV-based selection are depicted in Figure 6.75, 6.77 and 6.79. In general, these performance results are similar to those of the 3D-performance results: The ACV_{scaled} values are very similar independent of ts (Figure 6.75). The lowest range of ACV_{scaled} values is achieved for the configurations with $ts = 12$. The diversity values are increased by a rise of ts and are more similar for ts values equal or higher than 10 (Figure 6.77). The relative non-dominated solution quality is depicted in Figure 6.79 and reveals generally decreasing ACV_{rel} values by an increase of ts. Otherwise, the highest range of ACV_{rel} values as well as the highest

values in general are achieved for $ts = 15$. All configurations achieve outliers, which are in some cases remarkably higher than the maximal values. Compared to the 3D-performance results of the configurations with ACV-based selection, the decrease is more stringent in the case of the 3D-MOP (Figure 6.68). As the convergence values tend to be very low for the configurations with ACV-based selection and $ts = 10$, the diversity values are the second highest and the relative non-dominated solution quality is on a medium level compared to the other configurations, the setting $ts = 10$ is regarded as the advisable setting for this configuration. The comparison of these results to the 4D-performance results of the configurations with aggregate selection reveals that the convergence performance of both are absolutely comparable, but the diversity level is in general significantly lower in the case of the configurations with ACV-based selection. Otherwise, the relative non-dominated solution quality is significantly improved in the case of ACV-based selection compared to aggregate selection. The decision for the better selection strategy- aggregate or ACV-based selection - is a trade-off between higher diversity within the selected solutions or higher relative non-dominated solution quality.
The performance results of configurations with ACV-random selection and a variation of ts are presented in Figure 6.76, 6.78 and 6.80. The performance results are comparable to the corresponding 3D-results: The convergence performance does not reveal any regular performance variation by an increase of ts (Figure 6.76). The median values are equal for all configurations. The diversity performance also reveals no regularity in performance variation by an increase of ts (Figure 6.78). The configuration with $ts = 10$ tends to achieve the lowest range of diversity values among the different ts settings as well as the lowest ones in general. The relative non-dominated solution quality is nearly comparable (Figure 6.79). The tendency towards lowest performance results are achieved for $ts = 8, 10$ and 12. All configurations provide outliers indicating in some cases significantly higher performance values. Once again, since the random selection criterion is not discernible, the increased solution range of the tournament set does not influence the performance results in a predictable way and therefore does not influence the selection pressure. The comparison to the 4D-performance results of aggregate and ACV-based selection with the ts variation reveals that ACV-random selection performance lies somewhere in-between aggregate and ACV-based selection: Convergence and diversity results are remarkably higher than those of ACV-based selection and on the similar level than the aggregate selection performance. Otherwise, the relative non-dominated solution quality of the configurations with ACV-based selection is significantly lower than the performance of ACV-random selection.

The relative quality performance level of the configurations with ACV-random selection is lower compared to aggregate selection. As a consequence, the configurations with ACV-random selection are not able to outperform neither aggregate nor ACV-based selection.

6.2.3 Investigation of the Interdependency between ts and p_0

The following series of tests investigate the interdependence between ts and p_0 of the selection strategies in the case of the 3D- and 4D-MOP. The different configurations with various settings for ts and p_0 are examined according to convergence, diversity and relative non-dominated solutions quality to gain an insight into the issue of a performance regularity referring to these two selection parameters. The remaining configuration settings are the same as in the last series of tests. Theoretical considerations result in the hypothesis that a great tournament set in combination with a high selection probability result in a more discerning selection performance and therefore ensure a higher selection probability for high quality solutions.

Figure 6.81 depicts the 3D-performance results of aggregate selection strategy

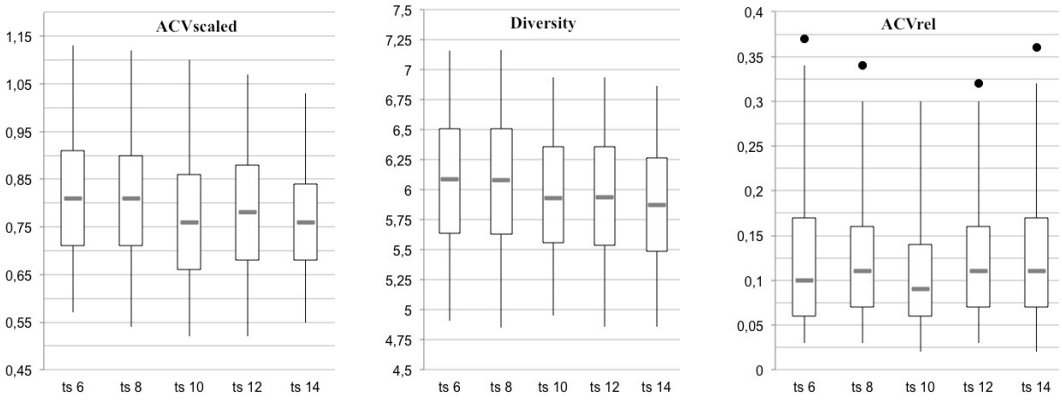

Fig. 6.81: 3D-results of aggregate selection with $p_0 = 20\%$ and a variation of ts.

with a selection probability of $p_0 = 20\%$. Convergence and diversity performance is comparable for $ts = 6$ and $ts = 8$, since the ACV_{scaled} and diversity values are of the same level. Furthermore, the convergence and the diversity metric values for $ts = 6, 8$ are the highest. An increase of ts results in a decrease of these metric values. The ACV_{rel} performance values, which assess the ACV values of the first front individuals in relation to the ACV values of the entire population, are the lowest for $ts = 10$. The configurations with $ts = 6$

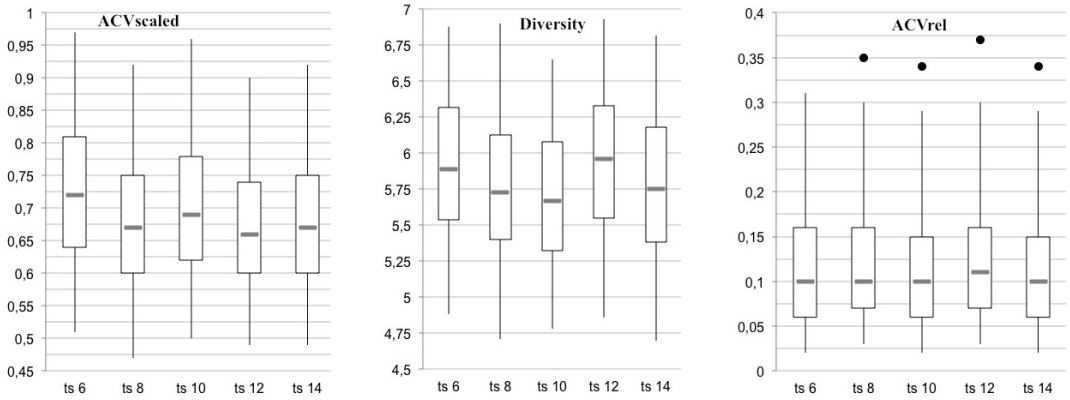

Fig. 6.82: 3D-results of aggregate selection with $p_0 = 40\%$ and a variation of ts.

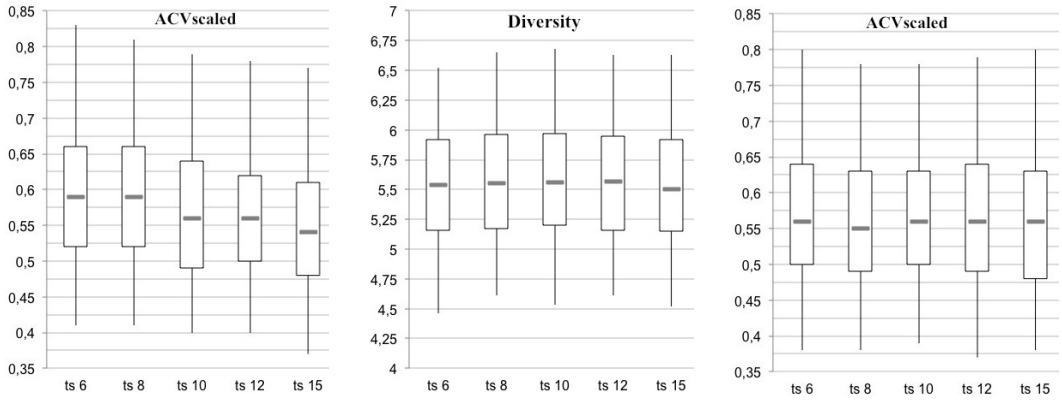

Fig. 6.83: 3D-results of aggregate selection with $p_0 = 60\%$ and a variation of ts.

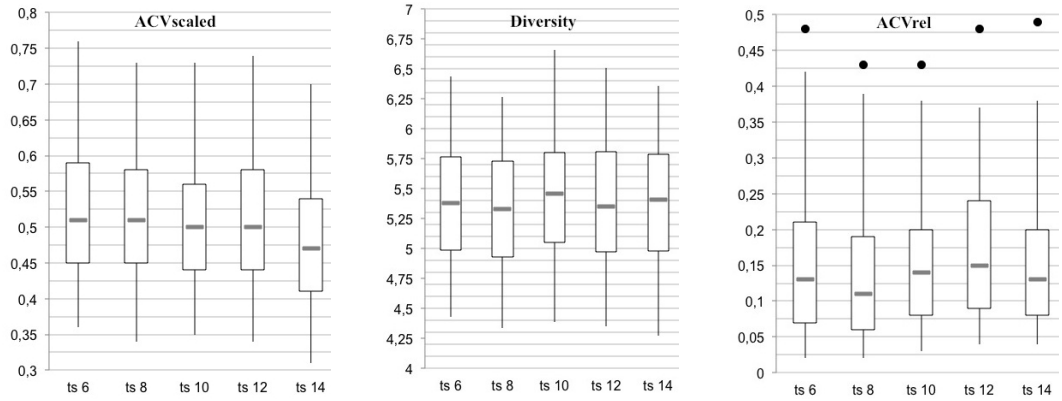

Fig. 6.84: 3D-results of aggregate selection with $p_0 = 70\%$ and a variation of ts.

and $ts = 14$ reveal a tendency towards highest ACV_{rel} values.
Figure 6.82 presents the performance results of the aggregate selection with $p_0 = 40\%$. There is no regularity observable regarding the convergence and diversity metric values by an increase of ts. The highest ACV_{scaled} and diversity values are achieved for $ts = 6$. From a global point of view, the convergence

and diversity performance are lower for higher ts values. The diversity results of $ts = 6$ and $ts = 12$ are comparable. The ACV_{rel} values are all of the same level for different ts settings and are therefore comparable.
The performance results of aggregate selection with $p_0 = 60\%$ are presented in Figure 6.83. Once again, convergence and diversity metric values are the highest for $ts = 6$ and 8. A further increase of ts results in a decrease of the ACV_{scaled} values and therefore in an improved convergence behavior. Diversity performance and ACV_{rel} values are comparable for all ts settings.
The performance results of aggregate selection with $p_0 = 70\%$ are depicted in Figure 6.84. The convergence behavior is improved by an increase of ts observable by the ACV_{scaled} values. Diversity values are slightly increased and comparable for $ts = 10$ and higher. There is no regularity observable regarding the ACV_{rel} values. The highest ACV_{rel} values are achieved for $ts = 12$. In general, the configurations with higher p_0 and ts values achieve a better performance in form of lower ACV_{scaled} values and higher or comparable diversity values. A significant improvement of the ACV_{rel} values is not observable by these settings.

Figure 6.85 depicts the 3D-performance results of ACV-based selection with

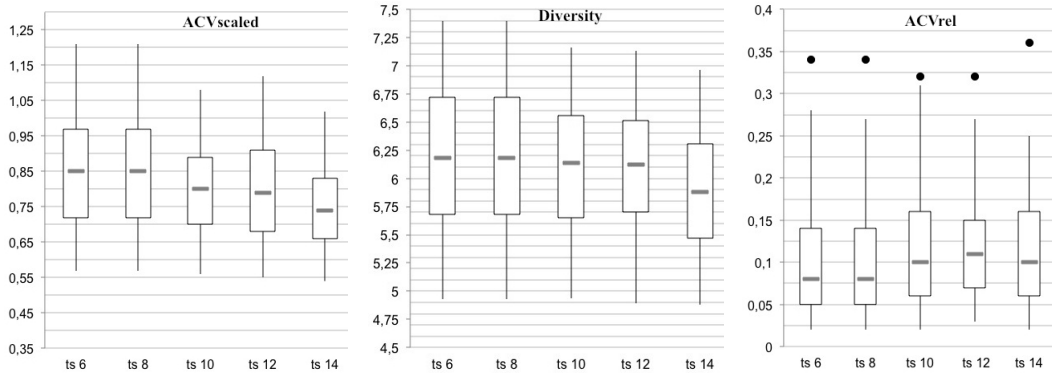

Fig. 6.85: 3D-results of ACV-based selection with $p_0 = 10\%$ and a variation of ts.

$p_0 = 10\%$. A tournament size lower than 10 results in similar performance results for ACV_{scaled}, diversity and ACV_{rel}. These performance results reveal the highest ACV_{scaled} and diversity values as well as the lowest ACV_{rel} values. The configurations with $ts = 10$ and higher achieve decreased convergence and diversity values, whereas the ACV_{rel} values are increased. Figure 6.86 presents the performance results of the configurations with $p_0 = 30\%$. The highest convergence and diversity values are achieved for $ts = 6$. A further increase of ts results in remarkably decreased convergence and diversity values. The performance values are generally lower for $p_0 = 30\%$ than for $p_0 = 10\%$ independent of ts. This is a potential consequence of the higher probability to

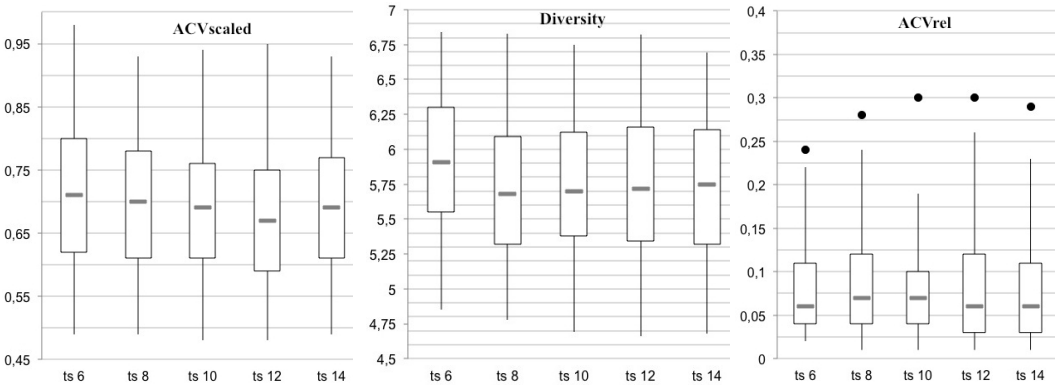

Fig. 6.86: 3D-results of ACV-based selection with $p_0 = 30\%$ and a variation of ts.

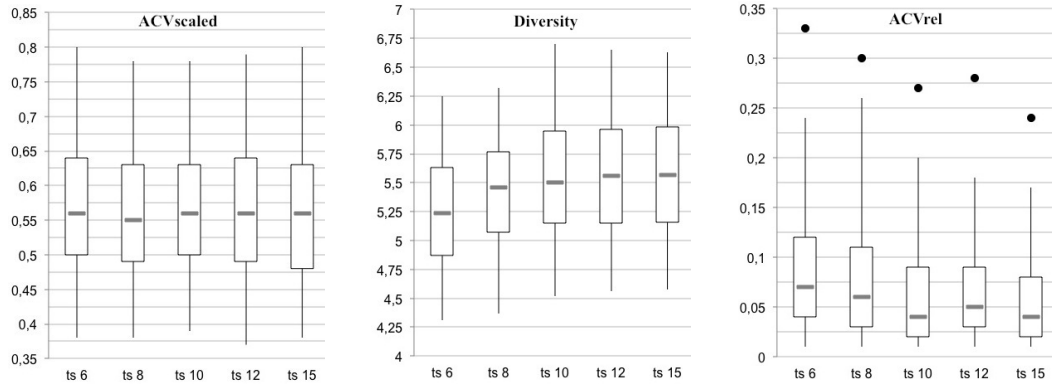

Fig. 6.87: 3D-results of ACV-based selection with $p_0 = 50\%$ and a variation of ts.

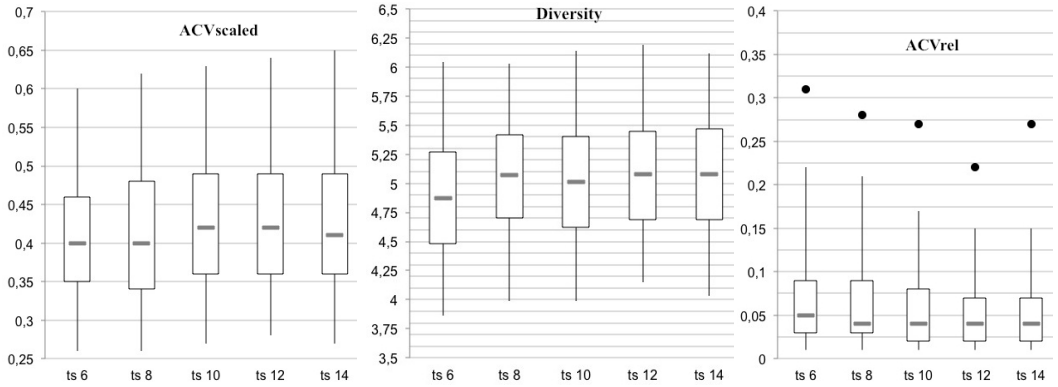

Fig. 6.88: 3D-results of ACV-based selection with $p_0 = 70\%$ and a variation of ts.

select individuals with the lowest ACV values. The performance results of the configurations with $p_0 = 50\%$ reveal a quite different behavior (Figure 6.87): The ACV_{scaled} values are indifferent by the variation of ts, whereas the diversity values are remarkably increased for $ts = 10$ and higher. The ACV_{rel} values are decreased for $ts = 10$ and higher. The performance results of the configurations with $p_0 = 70\%$ are completely different (Figure 6.88): The lo-

west convergence and diversity values are achieved for $ts = 6$. An increase of ts results in an increase of convergence and diversity. The ACV_{rel} values are decreased by an increase of ts.

These results (Figure 6.85 - Figure 6.88) do not reveal any interdependency between ts and p_0. Otherwise, the hypothesis mentioned in the introduction of this section is regarded as confirmed: Higher values for p_0 - or more concrete for $p_0 = 50\%$ and higher - in combination with high ts settings reveal a more discerning selection, since the ACV values of the non-dominated solutions are small relative to the ACV values of the entire population, observable by the generally lower ACV_{rel} values for $p_0 = 50\%$ or higher as well as $ts = 10$ and higher. Moreover, the configurations with the settings $p_0 = 30\%$ and 50% achieve lower or comparable ACV_{scaled} values with comparable or higher diversity values at the same time in the case of $ts = 10$ or higher.

Figure 6.89 depicts the 3D-performance results of ACV-random selection with

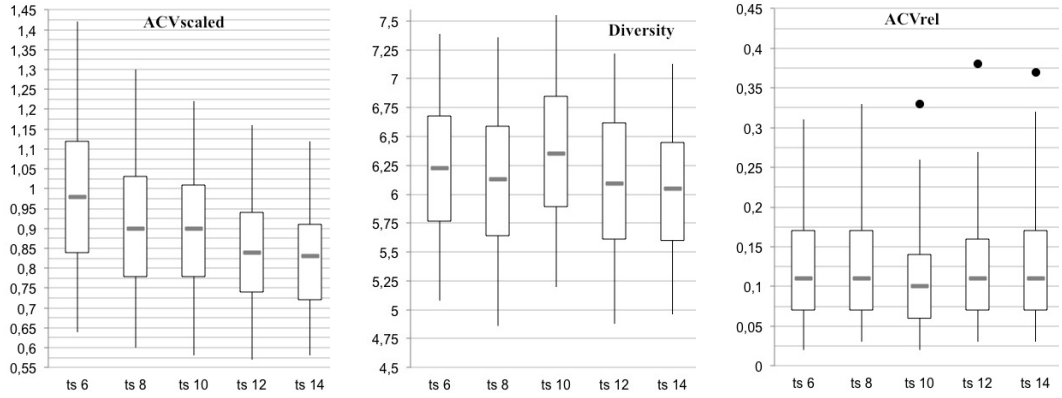

Fig. 6.89: 3D-results of ACV-random selection with $p_0 = 0\%$ (random selection) and a variation of ts.

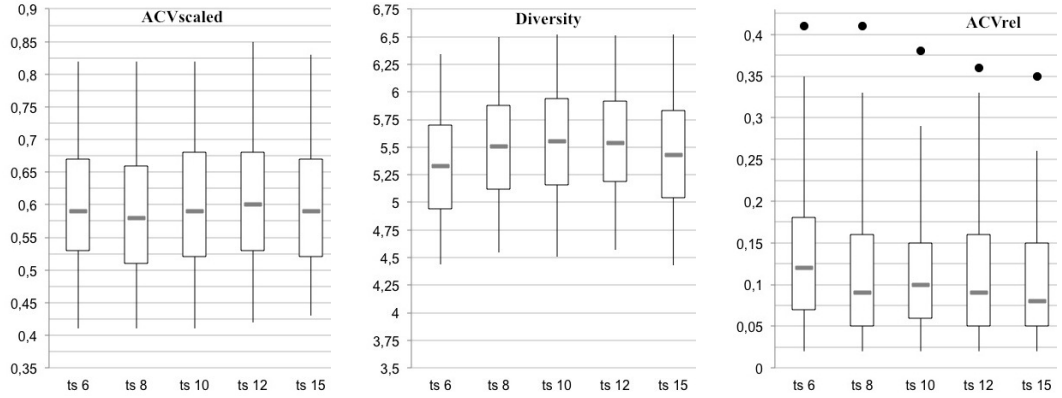

Fig. 6.90: 3D-results of ACV-random selection with $p_0 = 10\%$ and a variation of ts.

a selection probability of $p_0 = 0\%$ referring to the selection of individuals with

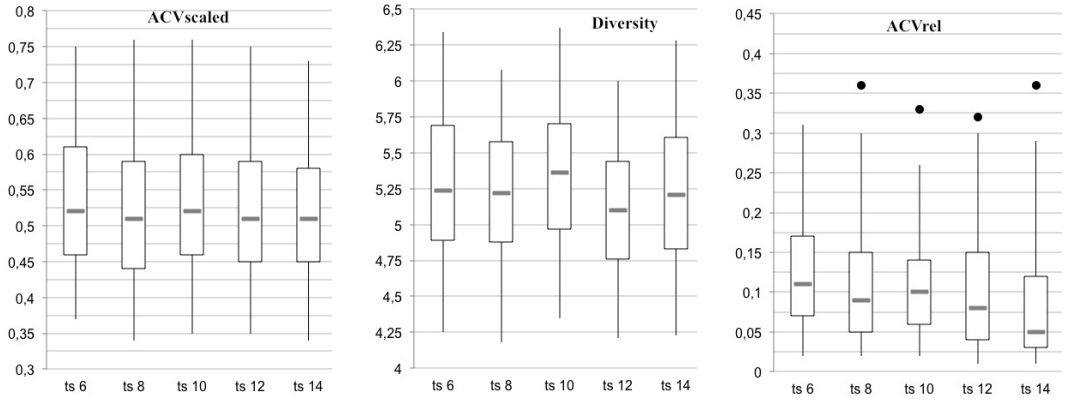

Fig. 6.91: 3D-results of ACV-random selection with $p_0 = 20\%$ and a variation of ts.

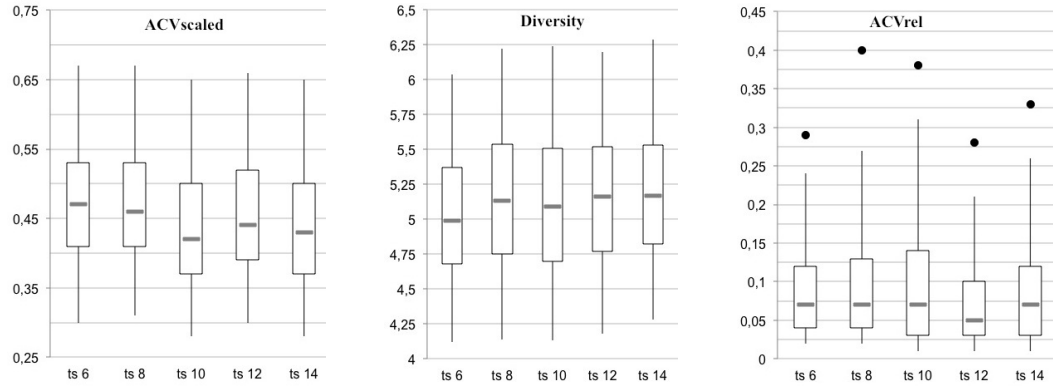

Fig. 6.92: 3D-results of ACV-random selection with $p_0 = 30\%$ and a variation of ts.

the lowest ACV_{scaled} value. This means that the individuals for the succeeding generation are sole randomly selected. The ACV_{scaled} values are decreased by an increase of ts. The diversity values do not reveal any regularity, the highest diversity values are achieved for $ts = 10$. The lowest ACV_{rel} values are achieved for $ts = 10$. The ACV_{rel} performance of the configurations with $ts = 6, 8$ and 14 are on the same level.

Figure 6.90 depicts the performance results of the ACV-random selection with $p_0 = 10\%$. There is no regularity observable regarding the ACV_{scaled} values by an increase of ts. The diversity is slightly increased by an increase of ts. The ACV_{rel} values are decreased by an increase of ts as the ACV_{rel} values are decreased. This indicates relatively small ACV values of the non-dominated solutions compared to the ACV values of the entire population

The performance results of aggregate selection with $p_0 = 20\%$ are presented in Figure 6.91. The worst convergence behavior is achieved with $ts = 6$. A further increase of ts results in a slight decrease of the ACV_{scaled} values and therefore in an optimized convergence performance. A regularity is not observable regarding the diversity values by an increase of ts. Otherwise, a decrease

of the ACV_{rel} values is observable by an increase of ts. Figure 6.92 presents the performance results of the configurations with $p_0 = 30\%$. The convergence is improved for $ts = 10$ and higher. Furthermore, the ACV_{scaled} values are comparable for ts values of 6 and 8. The lowest diversity values are achieved for $ts = 6$, those of $ts = 8$ and higher are remarkably improved and nearly comparable. The ACV_{rel} values increase slightly with an increase of ts from 6 to 10. The lowest ACV_{rel} values are achieved with $ts = 12$. A further increase to $ts = 14$ presents high ACV_{rel} values. In general, the hypothesis stated at the beginning of this section is regarded as confirmed: The configurations with $p_0 = 20\%$ and 30% achieve improved convergence behavior with increased or comparable diversity behavior at the same time in the case of $ts = 10$ and higher. Except from some outliers, the ACV_{rel} values are also increased for higher ts values.

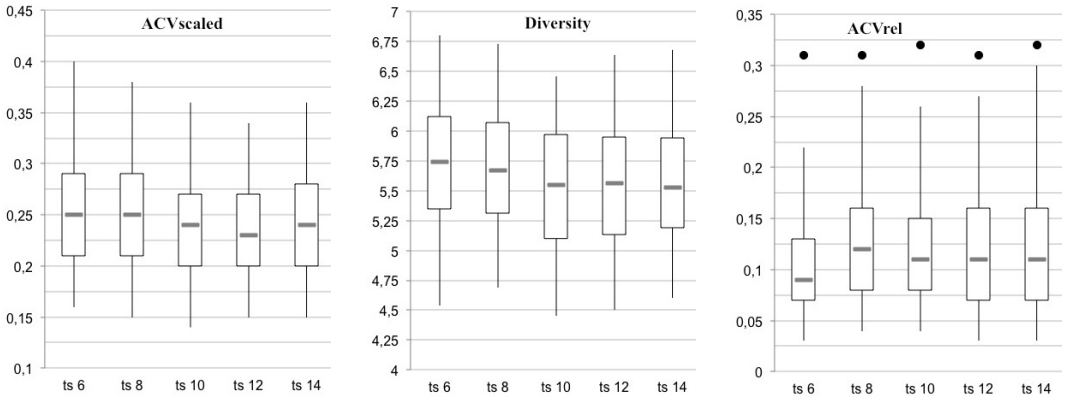

Fig. 6.93: 4D-results of aggregate selection with $p_0 = 20\%$ and a variation of ts.

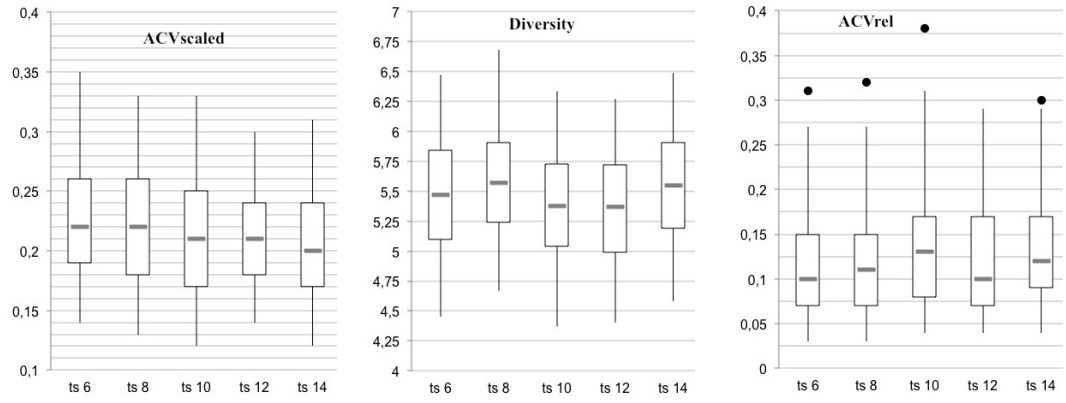

Fig. 6.94: 4D-results of aggregate selection with $p_0 = 40\%$ and a variation of ts.

In the following, the 4D-performance results of the selection strategies with a p_0 and ts variation are presented. Figure 6.93 depicts the performance results of aggregate selection with $p_0 = 20\%$, which indicates that 20% of the individuals

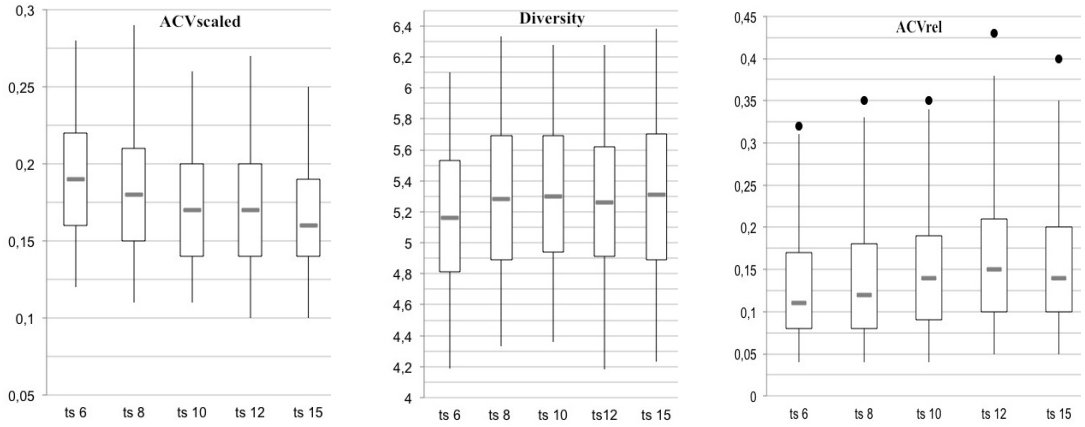

Fig. 6.95: 4D-results of aggregate selection with $p_0 = 60\%$ and a variation of ts.

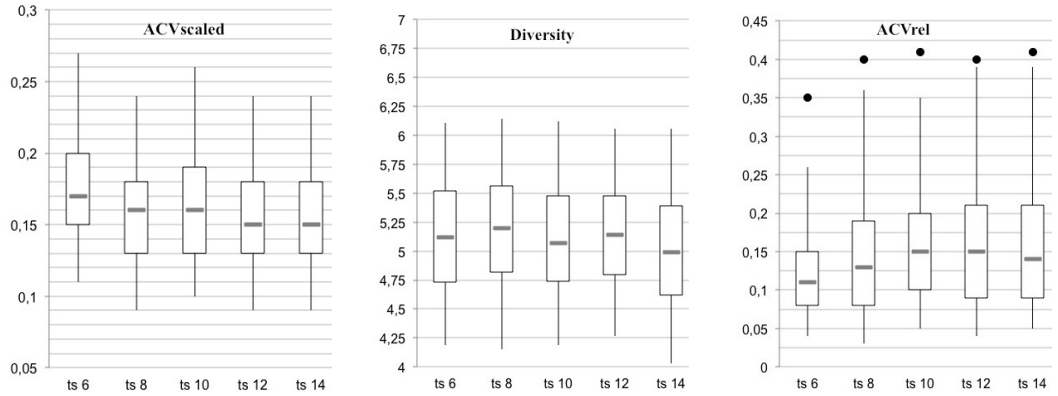

Fig. 6.96: 4D-results of aggregate selection with $p_0 = 70\%$ and a variation of ts.

are selected from the first front into the succeeding generations. The boxplots reveal similar convergence and diversity performance for $ts = 6$ and $ts = 8$ as well as the highest ACV_{scaled} and diversity values compared to the other ts settings. An increase of ts to 10 and higher reveals a tendency towards lower diversity and ACV_{scaled} values. The configurations with $ts = 12$ and $ts = 14$ achieve the highest ACV_{rel} values and therefore the highest ACV values of the non-dominated solutions relative to the ACV value of the corresponding entire population. Furthermore, the ACV_{rel} values reveal no regularity by a decrease of ts.

Figure 6.94 presents the 4D-performance results of aggregate selection with $p_0 = 40\%$. Once again, similar convergence and diversity results are achieved for $ts = 6$ and $ts = 8$. An increase of ts results in a decrease of the ACV_{scaled} value and therefore in an improved convergence performance. The variation of ts does not reveal any regularity regarding the diversity values. Highest diversity is achieved for $ts = 8$ and 14 and the lowest for $ts = 10$ and 12. The lowest ACV_{rel} values are achieved for $ts = 6$ and 8. An increase of ts results in an

increase of the ACV_{rel} values.
The 4D-performance results of the configurations with $p_0 = 60\%$ are presented in Figure 6.95. The highest ACV_{scaled} values are achieved for $ts = 6$. The increase of ts results in a significant decrease of the ACV_{scaled} values. The convergence performance for $ts = 10$ and $ts = 12$ are similar. The highest and also comparable diversity performance is achieved for $ts = 8, 10$ and 12. Regarding the ACV_{rel} values, an increase of ts results in an increase of the ACV_{rel} values.
The 4D-performance results of the last category of configurations with aggregate selection and $p_0 = 70\%$ are depicted in Figure 6.96. Once again, the configuration with $ts = 6$ reveals the highest ACV_{scaled} values. An increase of ts results in a general decrease of the ACV_{scaled} values and in a slight decrease of the diversity values. Furthermore, the increase of ts results in a continuous increase of the ACV_{rel} values. The ACV_{rel} are comparable for $ts = 12$ and 14. From a global point of view, the increase of the selection probability to select individuals from the first front results in a decrease of the ACV_{scaled} values with the increase of ts. The decrease of ACV_{scaled} values often correlates with the decrease of diversity. Furthermore, these performance results confirm the hypothesis that the individual selection from the first front does not guarantee the selection of the fittest individuals according to all objectives: The increase of ts results in an increase of ACV_{rel} values indicating no clear improvement of the non-dominated solution quality.

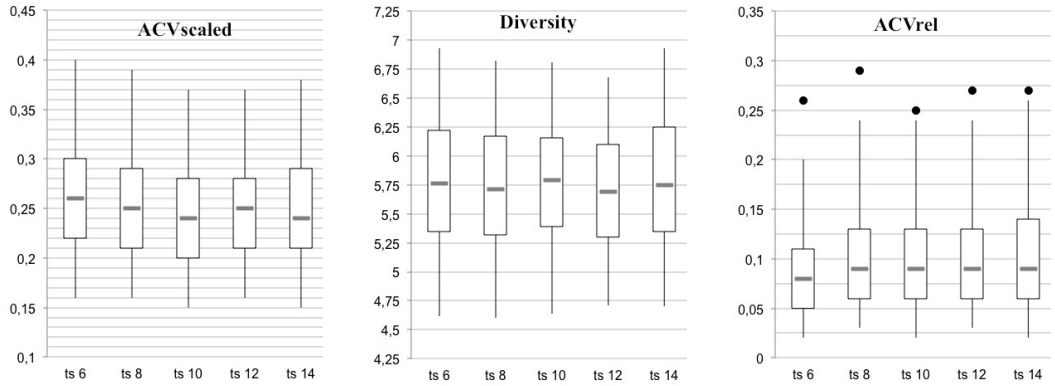

Fig. 6.97: 4D-results of ACV-based selection with $p_0 = 10\%$ and a variation of ts.

Figure 6.97 presents the 4D-performance results of the configurations with ACV-based selection and $p_0 = 10\%$. The highest ACV_{scaled} values are achieved for $ts = 6$ and a further increase of ts results in a slight decrease of the ACV_{scaled} values. The highest diversity values are achieved with $ts = 6$ and $ts = 14$. The diversity values for the other ts setting are slightly lower. The lowest ACV_{rel} values are achieved for $ts = 6$. An increase of ts results in an

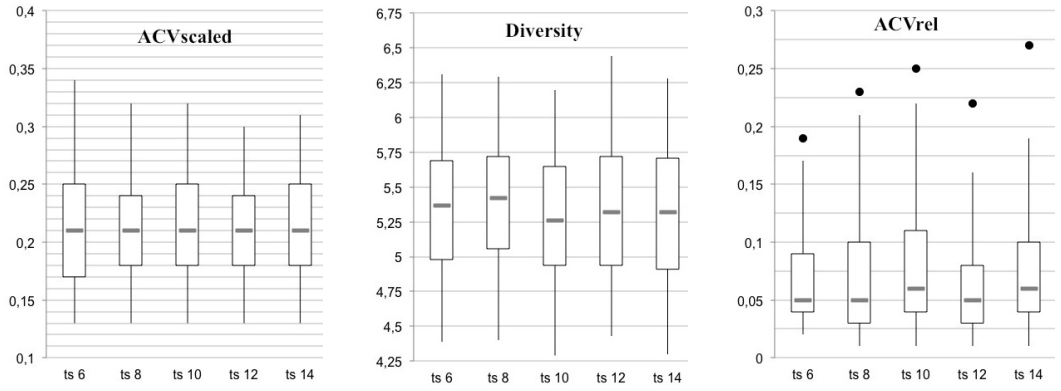

Fig. 6.98: 4D-results of ACV-based selection with $p_0 = 30\%$ and a variation of ts.

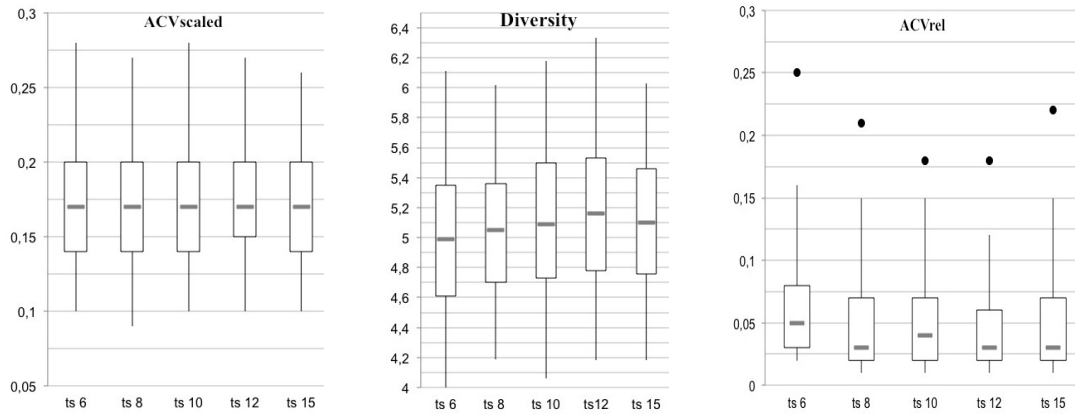

Fig. 6.99: 4D-results of ACV-based selection with $p_0 = 50\%$ and a variation of ts.

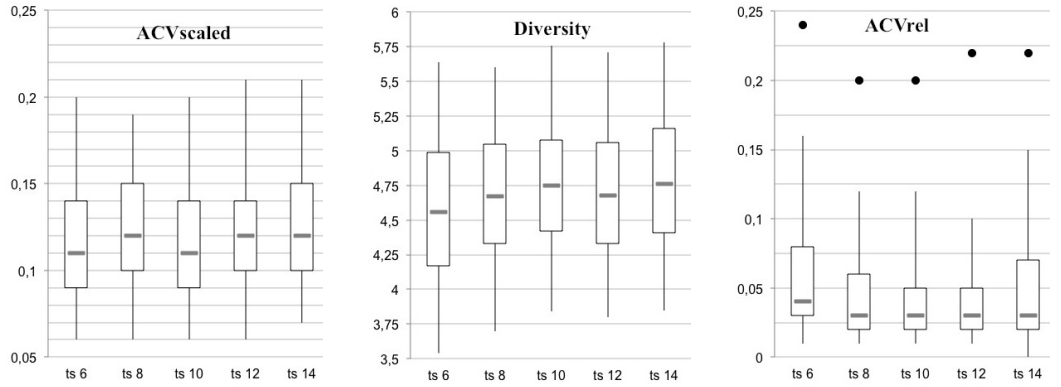

Fig. 6.100: 4D-results of ACV-based selection with $p_0 = 70\%$ and a variation of ts.

increase of the ACV_{rel} values. The ACV_{rel} values for $ts = 8$ and $ts = 14$ are comparable.

The 4D-performance results of the configurations with $p_0 = 30\%$ are depicted in Figure 6.98. In this case, there is no regularity observable regarding convergence and diversity by a variation of ts. The highest ACV_{rel} values are achieved for $ts = 6, 10$. Higher ts values reveal a tendency towards lower ACV_{rel} values

indicating improved non-dominated solution quality relative to the convergence performance of the entire population.
The 4D-performance results of the configurations with $p_0 = 50\%$ (Figure 6.99) reveal comparable ACV_{scaled} values and the increase of ts results in a remarkable increase of the diversity values. The increase of ts results in a decrease of the ACV_{rel} values and therefore once again in improved non-dominated solutions.
The 4D-performance results of the configurations with $p_0 = 50\%$ (Figure 6.100) reveal comparable ACV_{scaled} values for $ts = 6, 10$ and 12. The highest ACV_{scaled} values are achieved for $ts = 8, 14$. A slight increase of the diversity values is observable by an increase of ts. The ACV_{rel} values are significantly decreased by an increase of ts, except from the outlier $ts = 14$.
In general, the increase of the selection probability up to 50% and higher reveals improved ACV_{scaled} values, usually independent of ts at the same time. The diversity is increased by higher ts values up to 10 and higher. Furthermore, $ts \geq 10$ reveals a more discerning selection. Concluding, higher ts values in combination with higher selection probabilities support a more discerning selection.

Figure 6.101 presents the 4D-performance results of the configurations with

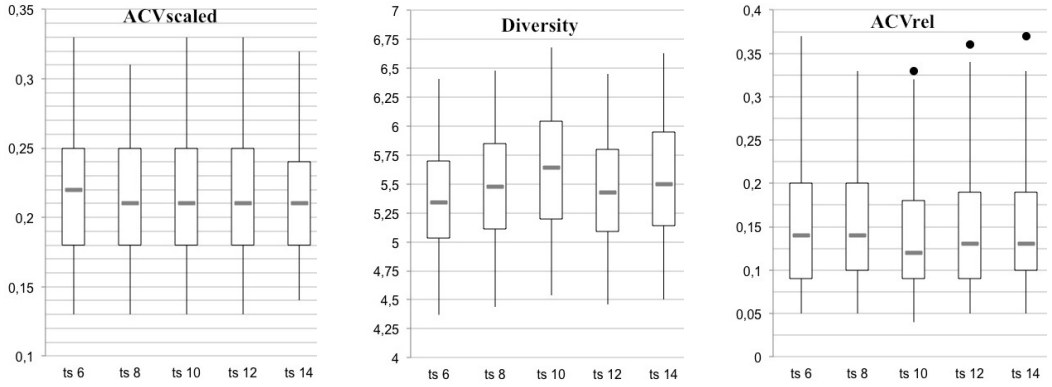

Fig. 6.101: 4D-results of ACV-random selection with $p_0 = 0\%$ (random selection) and a variation of ts.

ACV-random selection and a selection probability of $p_0 = 0\%$, the percentage of selecting individuals into the succeeding generation according to the lowest ACV value. The ACV_{scaled} values are highly comparable for all ts settings. Furthermore, there is no regularity observable regarding the diversity performance and the increase of ts. The highest diversity values are achieved for $ts = 10$ and 14. The ACV_{rel} values are the highest for $ts = 6$ and 8, a further increase of ts results in a general decrease of ACV_{rel} values.
The performance results of the configurations with ACV-random selection and

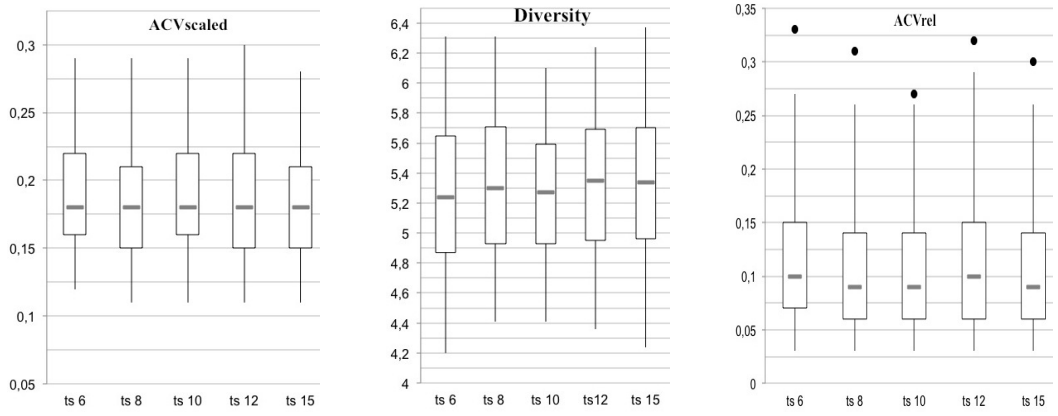

Fig. 6.102: 4D-results of ACV-random selection with $p_0 = 10\%$ and a variation of ts.

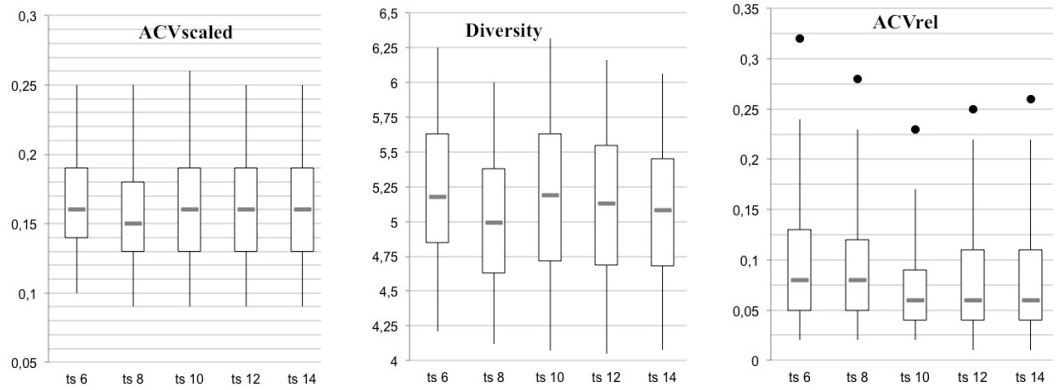

Fig. 6.103: 4D-results of ACV-random selection with $p_0 = 20\%$ and a variation of ts.

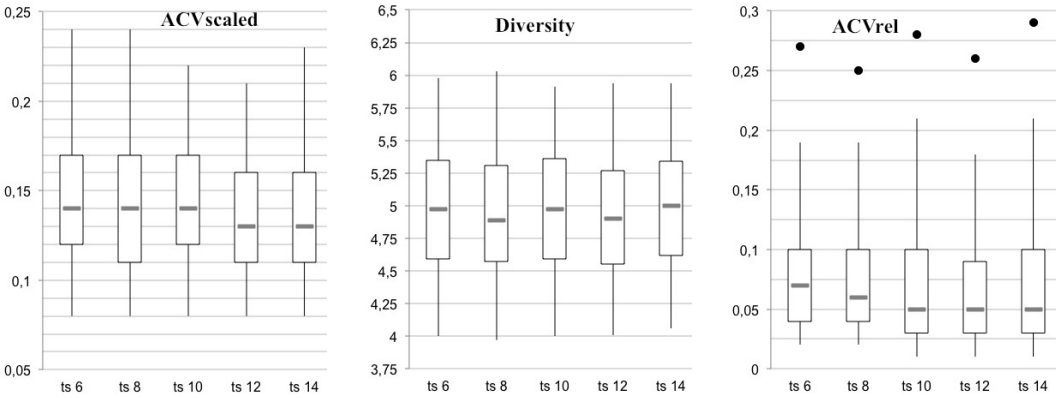

Fig. 6.104: 4D-results of ACV-random selection with $p_0 = 30\%$ and a variation of ts.

$p_0 = 10\%$ (Figure 6.102) reveal an oscillating convergence and diversity performance by an increase of ts. The ACV_{rel} values are nearly comparable for all ts settings. The performance results of a further increase of p_0 to 20% reveal once again highly comparable ACV_{scaled} values (Figure 6.103). The lowest ACV_{scaled} values are achieved for $ts = 8$. The highest diversity values are achieved for

$ts = 6$ and 10. A further increase of ts results in a continuous decrease of the diversity values. The highest ACV_{rel} values are achieved for $ts = 6$. An increase of ts up to 10 and higher results in lower ACV_{rel} values.
The performance results of the configurations with $p_0 = 30\%$ reveal once again highly comparable ACV_{scaled} values for $ts = 6$ and 8 (Figure 6.104). A general decrease of the ACV_{scaled} values is achieved for higher values $ts = 12$ and 14. The diversity values are nearly comparable and no regularity is observable by an increase of ts. Also the ACV_{rel} values are comparable for all ts settings.
In general, there is no regularity observable regarding convergence, diversity or relative non-dominated solution quality by a variation of ts and p_0. In the case of this relatively low probability, the ACV indicator as selection criterion is not able to reveal the more discerning selection effect. Therefore, the random selection is pre-dominant and results in comparable or oscillating metric values by a variation of ts.

6.2.4 Discussion

The performance results of the test series focused on selection strategies are further summarized regarding the influence of the selection parameters on the performance in the case of the 3D- and 4D-MOP. Furthermore, the search performance of different selection strategies is discussed in a qualitative way. The preferable selection strategy including advisable parameter settings is assessed under the aspects of a good-convergence-diversity balance, an optimal relative non-dominated solution quality and computational complexity arising in each iteration step. In general, the experimental results reveal that the advisable parameter settings for each selection strategy are equal and independent of the problem dimension. However, the parameters 'selection probability' and 'tournament size' have a different influence on performance, depending on the selection strategy. This strategy-specific influence is the same in the case of the 3D- and 4D-MOP. The increase of the selection probability, which is responsible for the percentage of selecting individuals either from the first front or the individuals with the lowest ACV values, results in improved convergence performance, mostly at the cost of diversity. In general, the experimental results reveal that the best performance results are achieved in the case of the three selection strategies with $ts = 10$ and higher. The worst performance results are achieved for $ts = 6$ and acceptable results for $ts = 8$ in the case of all three selection strategies as well as the 3D- and 4D-MOP.
The result matrix presents the performance of the three selection strategies

		Aggregate select. p=60%, ts=10	ACV-based select. p=50%, ts=10	ACV-random select. p=10%, ts=10
3D-MOP	ACV_{scaled}	[0.53, 0.62]	[0.53, 0.61]	[0.55, 0.65]
	Diversity	[5.33, 5.83]	[5.30, 5.82]	[5.31, 5.82]
	ACV_{rel}	[0.10, 0.18]	[0.04,0.10]	[0.08, 0.16]
4D-MOP	ACV_{scaled}	[0.15, 0.19]	[0.15, 0.19]	[0,17, 0.21]
	Diversity	[5.08, 5.57]	[4.88, 5.35]	[5.04, 5.51]
	ACV_{rel}	[0.11, 0.19]	[0.03,0.08]	[0.08, 0.14]

Table 6.4: Results matrix of convergence, diversity and relative non-dominated solution quality achieved by VONSEA configurations with various selection strategies in the case of the 3D- and 4D-MOP.
Backgrounds: the optimal indicator result (green), medium performance (blue), the worst indicator result (red).

with advisable parameter settings for the 3D- and 4D-MOP (Table 6.4). The presented intervals are 2σ-confidence limits of the means of the indicator results achieved by different configurations. The optimal indicator results are each highlighted in green, the worst one in red and indicator values of a medium performance are in blue. In the case of the 3D-MOP, there is a clear advantage of the ACV-based selection observable compared to aggregate selection: The convergence-diversity balance is similarly good in both cases, but the non-dominated solution quality is significantly improved compared to the configuration with aggregate selection. This effect is the initial motivation for ACV-based selection. The aim of ACV-based selection is to overcome the disadvantage of aggregate selection, which does not guarantee the selection of the fittest individual of the tournament set according to all objectives by the random selection of a first front individual.
In the case of the 4D-MOP, the advantage of the ACV-based selection is not as obvious as in the case of the 3D-MOP. Configurations with ACV-based selection generally achieve better convergence and a significantly better relative non-dominated solution quality relative to the convergence performance of the entire population than the configurations with aggregate selection, but the diversity is remarkably lower. The performance of configurations with ACV-random selection usually lies in-between those of aggregate and ACV-based selection and is therefore not able to outperform these selection strategies. Furthermore, ACV-random selection allows no predictable performance results by a variation of the parameter p_0. These ACV-random selection results reveal that fitness-proportionate selection used as selection criterion has a particular importance in the area of peptide - or more generally - biochemical optimization.
Table 6.5 presents the result matrix of the three selection strategies with advi-

	ts=6	ts=8	ts=10	ts=12	ts=15
AGG select. 60%	[0.55, 0.64] [5.27, 5.81] [0.09, 0.17]	[0.55, 0.64] [5.33, 5.83] [0.09, 0.17]	[0.53, 0.62] [5.33, 5.83] [0.10, 0.18]	[0.52, 0.61] [5.29;5.86] [0.11;0.21]	[0.50, 0.59] [5.29;5.80] [0.09;0.16]
ACV-based 50%	[0.52, 0,62] [5.01, 5.51] [0.05, 0.13]	[0.52, 0.61] [5.20, 5.62] [0.05, 0.12]	[0.53, 0.61] [5.30, 5.82] [0.04,0.10]	[0.52, 0.61] [5.31, 5.85] [0.04, 0.11]	[0,52, 0,61] [5.34, 5.84] [0.04, 0.09]
ACV-random 10%	[0.56, 0,65] [5.11, 5.58] [0.09, 0.19]	[0.54, 0,63] [5.26, 5.76] [0.08, 0.17]	[0.55, 0.65] [5.31, 5.82] [0.08, 0.16]	[0.56, 0.66] [5.30, 5.81] [0.08, 0.16]	[0.56, 0.65] [5.19, 5.72] [0.07, 0.15]

Table 6.5: Results matrix of convergence, diversity and relative non-dominated solution quality achieved by configurations with various selection strategies, advisable p_0 settings and a variation of ts in the case of the 3D-MOP.
Backgrounds: the optimal indicator result (green), medium performance (blue), the worst indicator result (red).

	ts=6	ts=8	ts=10	ts=12	ts=15
AGG select. 60%	[0.17, 0.21] [4.93;5.42] [0.10, 0.16]	[0.16, 0.20] [5.03, 5.57] [0.10, 0.18]	[0.15, 0.19] [5.08, 5.57] [0.11, 0.19]	[0.15, 0.19] [5.01, 5.50] [0.12, 0.22]	[0.15, 0.19] [5.05, 5.58] [0.12, 0.21]
ACV-based 50%	[0.15, 0.19] [4.57, 5.24] [0.04. 0.10]	[0.15, 0.19] [4.84, 5.27] [0.03, 0.08]	[0.15, 0.19] [4.86, 5.37] [0.03, 0.08]	[0.16, 0.20] [4.92, 5.44] [0.03, 0.07]	[0.15, 0.19] [4.88, 5.35] [0.03, 0.08]
ACV-random 10%	[0.17, 0.21] [5.0, 5.52] [0.08, 0.16]	[0.17, 0.21] [5.08, 5.57] [0.08, 0.15]	[0,17, 0.21] [5.04, 5.51] [0.08, 0.14]	[0.17, 0.21] [5.08, 5.56] [0.09, 0.15]	[0.16, 0.20] [5.07, 5.60] [0.08, 0.14]

Table 6.6: Results matrix of convergence, diversity and relative non-dominated solution quality achieved by configurations with various selection strategies, advisable p_0 settings and a variation of ts in the case of the 4D-MOP.
Backgrounds: the optimal indicator result (green), medium performance (blue), the worst indicator result (red).

sable p_0 settings and a variation of ts in the case of the 3D-MOP. The optimal indicator results are each highlighted in green, the worst one in red and indicator values of a medium performance are in blue. The upper intervals in each box are 2σ-confidence limits of the means of ACV_{scaled} values, the second intervals are diversity 2σ-confidence limits and the intervals given below are 2σ-confidence limits of ACV_{rel} values. The result matrix of the 3D-MOP performance reveals a superficial impact of ts variation on convergence, diversity and relative non-dominated solution quality in the case of the three selection strategies: The worst indicator results are mainly achieved for $ts = 6$, an increase of ts results in an improvement or a stagnation of the performance results in most cases. More precisely, the influence of ts on the indicator results is different for the selection strategies and also different in the case of different p_0 settings. The configurations with aggregate selection and a high ts value result in improving convergence behavior but worse performance results regarding

the non-dominated solution quality relative to the convergence results of the entire population. The aggregate selection chooses the individuals from the first front of a different sized tournament set with the advisable probability of $p_0 = 60\%$. The rising size of the tournament set results in more sophisticated first front individuals, but the random selection of one individuals from the first front does not guarantee the selection of the highest qualified individual according to all objectives. The configurations with ACV-based selection, a probability value of $p_0 = 50\%$ and a variation of ts reveal highly comparable convergence results. Furthermore, the relative non-dominated solution quality is slightly improved by an increase of ts and are the best results in general compared to the other ACV_{rel} results of aggregate or ACV-random selection. High relative non-dominated solution quality is referred to the fact that 50% of the individuals are selected based on the lowest ACV_{scaled} value. The convergence performance is only slightly decreased compared to the results of aggregate selection. The configurations with ACV-random selection reveal no regularity regarding the increase of ts. The relative non-dominated solution quality is nearly comparable for all ts settings and is of medium performance compared to the other configurations. Diversity and convergence performance are oscillating. This is a consequence of the fact that the selection process is highly dominated by a random selection of individuals into the succeeding generation, since only 10% of the individuals are selected according to the lowest ACV_{scaled} value.

Table 6.6 presents the corresponding confidence limits of the configurations with the three selection strategies, advisable p_0 settings and a variation of ts in the case of the 4D-MOP. The conclusions of this results matrix are the same as in the case of the 3D-MOP: Higher ts values result in better convergence performance in the case of aggregate selection, but the relative non-dominated solution quality is the worst compared to the other results. The configurations with ACV-based selection provide the best convergence performance independent of ts settings, at the cost of diversity. The relative non-dominated solution quality improvement is independent of ts settings. In the case of ACV-random selection, the relative non-dominated solution quality is on a medium level for all ts settings, while convergence and diversity performance are once again oscillating.

As a consequence, a ts setting of approximately 10 is regarded as an advisable setting for all selection strategies with respect to a good and reliable convergence-diversity balance. A higher number for ts results in an increasing computational complexity.

6.3 Number of Parents for Recombination

The test series documented in this section investigate the influence of the parent number for recombination on convergence and especially on diversity performance. Furthermore, this influence is investigated in combination with a population size (ps) variation to gain an insight into the interdependence between the parent number for recombination and the population size. The previous experiments revealed that the recombination has only a low influence on the algorithm performance. This allows the hypothesis that the variation of the parent number has no considerable influence in general on the performance. The influence of the multi-parent recombination is investigated for LiDeRP and ExpoDeRP, which themselves have been revealed as advisable recombination operators in Section 6.1. The further algorithm parameters and components are set as follows: The default parent number for recombination is $pn = 3$, BS_{adapt} is used as mutation and ACV-based selection with the parameters $p_0 = 50\%$ and $ts = 10$. These components have proven to compose an advisable combination regarding a robust performance. However, the ACV-based selection reveals superior performance in the case of the 3D-MOP, but the question of an advisable selection strategy for the 4D-MOP is a trade-off between lower diversity and high non-dominated solution quality or high diversity at the cost of non-dominated solution quality. For a better comparison of the 3D- as well as 4D-performance results of the test series, ACV-based selection is also used as selection strategy in the case of the 4D-MOP.

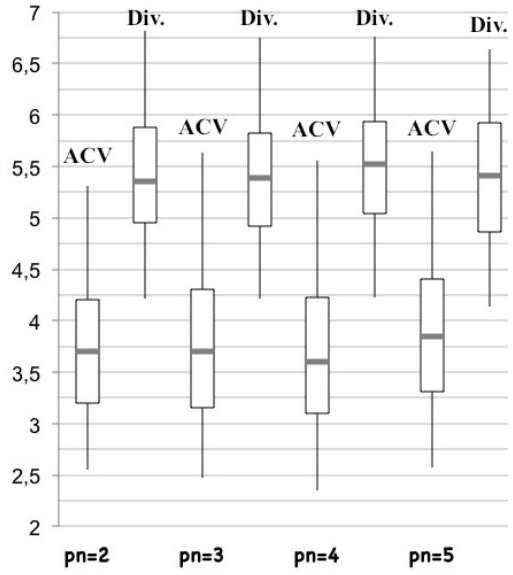

Fig. 6.105: Performance of multi-parent LiDeRP with $ps = 60$, (3D-MOP)

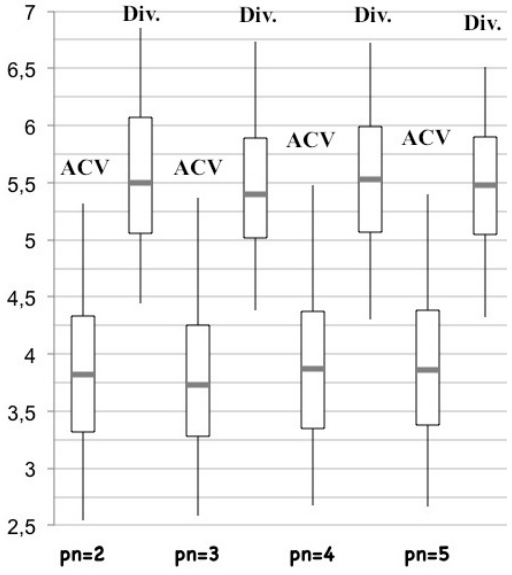

Fig. 6.106: Performance of multi-parent LiDeRP with $ps = 80$, (3D-MOP)

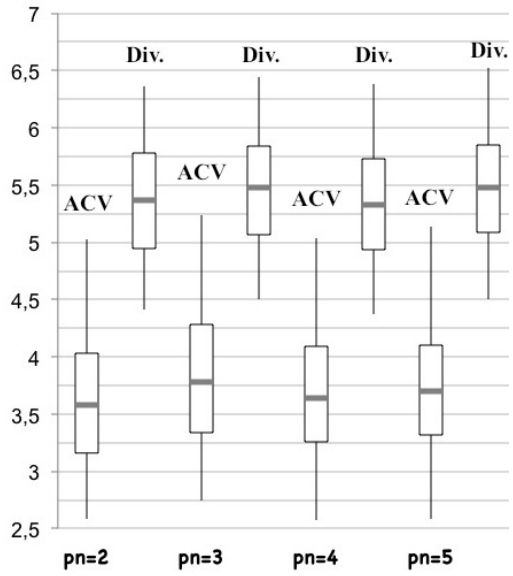

Fig. 6.107: Performance of multi-parent LiDeRP with $ps = 100$, (3D-MOP)

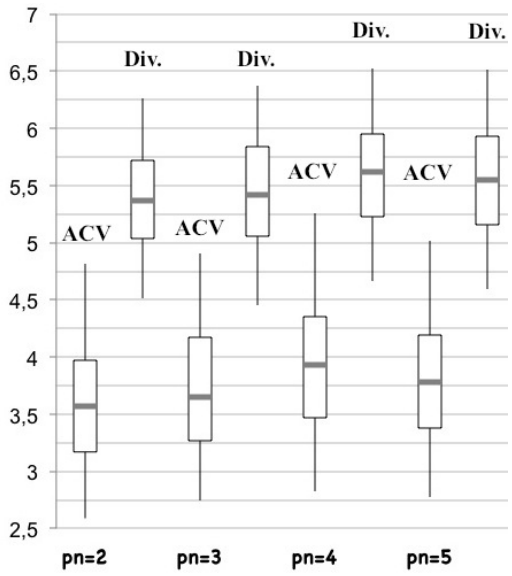

Fig. 6.108: Performance of multi-parent LiDeRP with $ps = 120$, (3D-MOP)

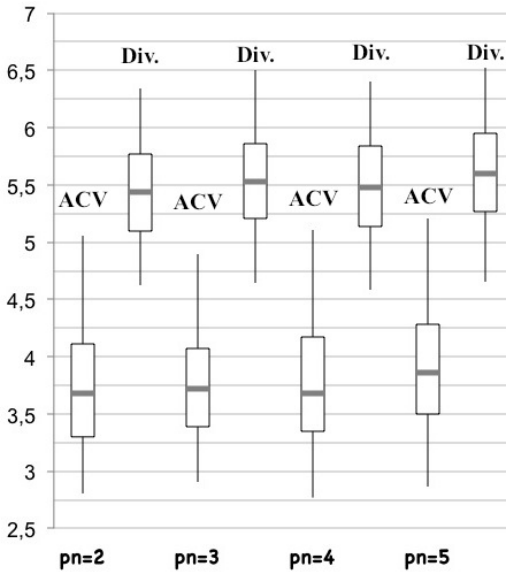

Fig. 6.109: Performance of multi-parent LiDeRP with $ps = 140$, (3D-MOP)

6.3.1 LiDeRP: Multi-parent Variation

In the following, the results of the multi-parent recombination with LiDeRP as recombination operator are presented. Firstly, the experimental results of the 3D-MOP are discussed: Figure 6.107 depicts the performance results of the configurations with the default population size of 100 and a variation of the

parent number in the range from 2 to 5. The performance results do not reveal any regularity by the increase of the parent number. The tendency towards the highest metric values is achieved by the choice of three parents for recombination followed by the performance results of the configuration with five parents. The best convergence results are achieved by two parents for recombination, at the cost of diversity. Figure 6.105 and 6.106 present the performance results of the configurations with a lower population size of 60 and 80. The decrease of the population size produces the same results by a variation of the parent number, compared to the population size of 100: In general, there is no regularity observable regarding the metric results by an increase of the parent number. The tendency towards the lowest metric results is achieved by three parents in the case of a population size of 80, the highest is achieved by two parents (Figure 6.106). The convergence performance is equal in the case of four and five parents for recombination. The performance results of the configuration with a population size of 60 are more equal and in-differentiable (Figure 6.105): The tendency towards the highest diversity results is achieved by four parents with the tendency towards acceptable convergence results. The highest ACV values are achieved with five parents, revealing the widest range of diversity values at the same time. Figure 6.108 and 6.109 present the performance results of the configurations with the higher population size of 120 and 140. In general, the metric values are increased by higher parent number $pn = 4$ and 5 for both configurations (population sizes 120 and 140) compared to the configuration with a population size of 100. The configurations with a population size of 120 reveal an increase of the metric values with an increase of the parent number from $pn = 2$ up to $pn = 4$ and decrease afterwards for $pn = 5$ (Figure 6.108). The configurations with a population size of 140 reveal a tendency towards the lowest metric values for two parents and the highest for five parents. The performance results are generally more comparable than in the case of the configurations with lower population sizes.

The performance results of a parent number variation achieved by configurations with different population sizes in the case of the 4D-MOP are depicted in the figures 6.110 to 6.114. The performance results are further discussed in comparison to the corresponding results in the case of the 3D-MOP. The configurations with the default population size of 100 reveal an increase of the metric values by an increase of the parent number up to $pn = 4$ (Figure 6.112). The results of $pn = 5$ are on the same level as the configuration with $pn = 3$. A decrease of the population size to 80 results in similar performance compared to the configurations with $ps = 100$ (Figure 6.111). The ACV values of $pn = 3$ and $pn = 5$ achieve the highest ACV values compared to the other configu-

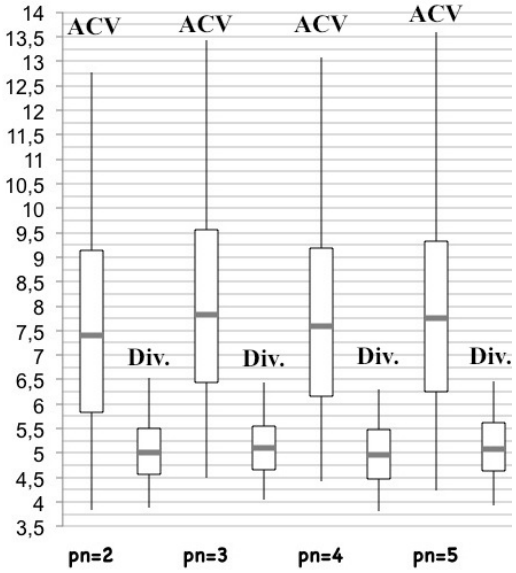

Fig. 6.110: Performance of multi-parent LiDeRP with $ps = 60$, (4D-MOP)

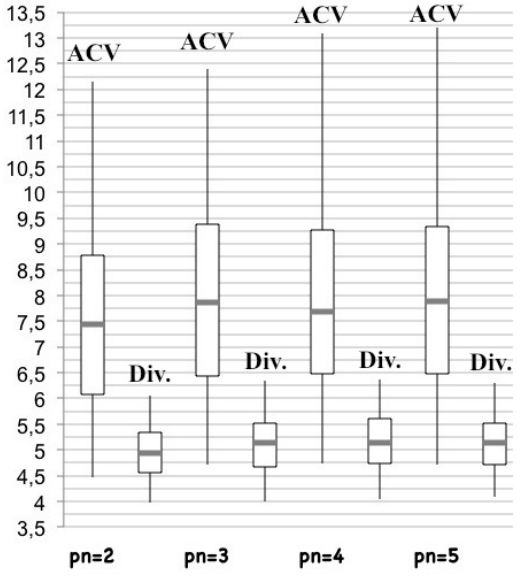

Fig. 6.111: Performance of multi-parent LiDeRP with $ps = 80$, (4D-MOP)

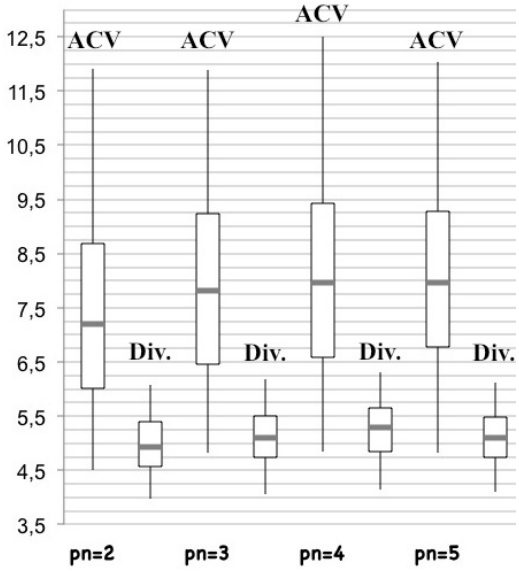

Fig. 6.112: Performance of multi-parent LiDeRP with $ps = 100$, (4D-MOP)

rations. A further decrease of ps to 60 reveals similar performance compared to the performance results of the 3D-MOP in the sense that no regularity is observable by an increase of the parent number. Once again, the configuration with $pn = 3$ achieves the highest performance results. An increase of ps to 120 and 140 results in an increase of the diversity values by an increase of pn (Figure 6.113, Figure 6.114). Furthermore, in both cases the ACV values are comparable for $pn = 2$ and $pn = 4$. The highest metric values are achieved for the configuration with $pn = 5$. The comparison of these observations compared

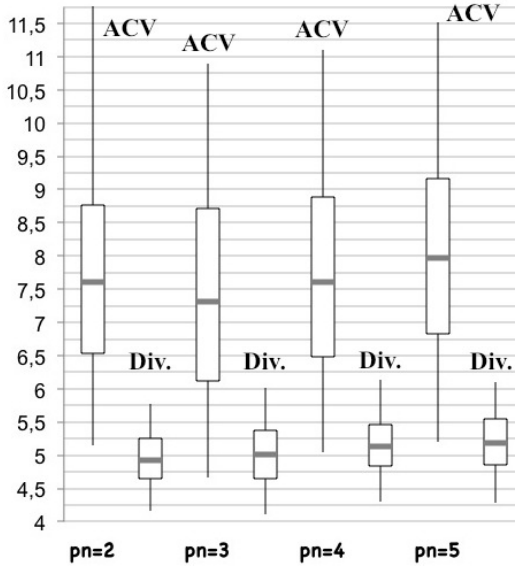

Fig. 6.113: Performance of multi-parent LiDeRP with $ps = 120$, (4D-MOP)

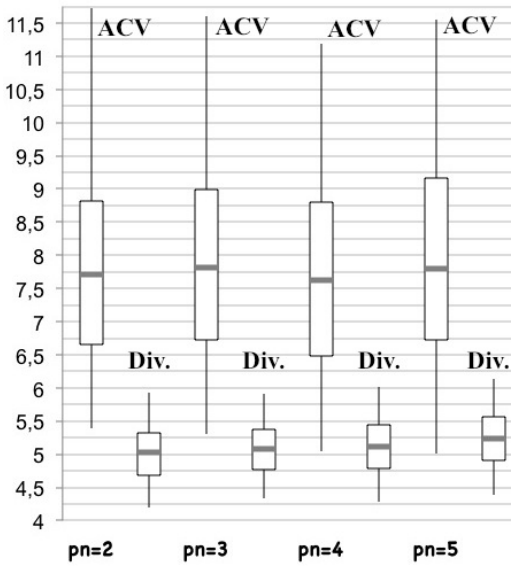

Fig. 6.114: Performance of multi-parent LiDeRP with $ps = 140$, (4D-MOP)

to the performance results of the 3D-MOP reveals a general influence of the parent number on the performance results that is mostly independent of the population size: As theoretically expected, the increase of the parent number results in an increase of the diversity values. But this higher diversity of the genetic material goes along with an increase of the ACV values at the same time and therefore results in an overall worse convergence performance. In the cases of $ps = 60, 80$ and $ps = 140$, the ACV values of the configurations with $pn = 3$ have a tendency towards exceptionally high values compared to the results of the $pn = 2$ and $pn = 4$.

6.3.2 ExpoDeRP: Multi-parent Recombination

In the following, the experimental results of the multi-parent recombination with ExpoDeRP are presented. Firstly, the results of the 3D-MOP are discussed. Generally, ExpoDeRP provides a lower number of recombination points from the second generation on caused by the decreasing exponential function. From a theoretical point of view, a lower parent number for the ExpoDeRP recombination is therefore more motif-maintained from the second generation on, referring to the genetic material, than a lower parent number in combination with LiDeRP.
Figure 6.117 depicts the performance results of the configurations with the default population size of 100 and a variation of the parent number in the ran-

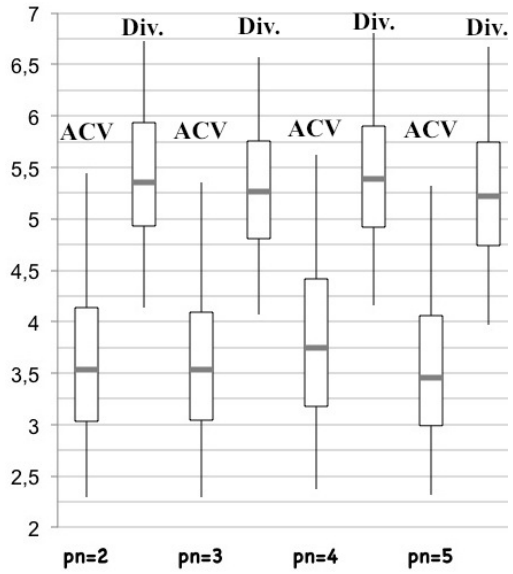

Fig. 6.115: Performance of multi-parent ExpoDeRP with $ps = 60$, (3D-MOP)

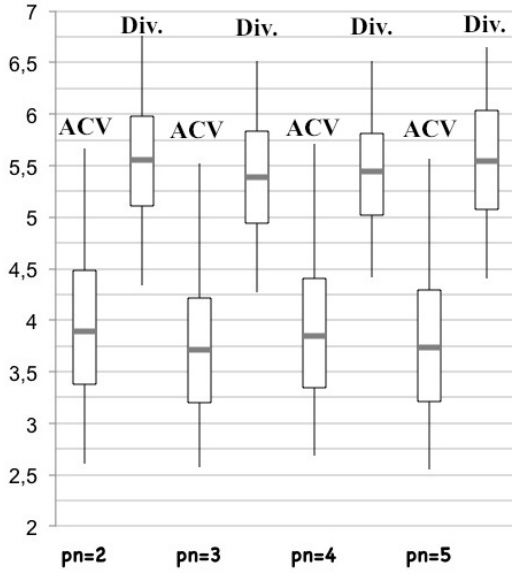

Fig. 6.116: Performance of multi-parent ExpoDeRP with $ps = 80$, (3D-MOP)

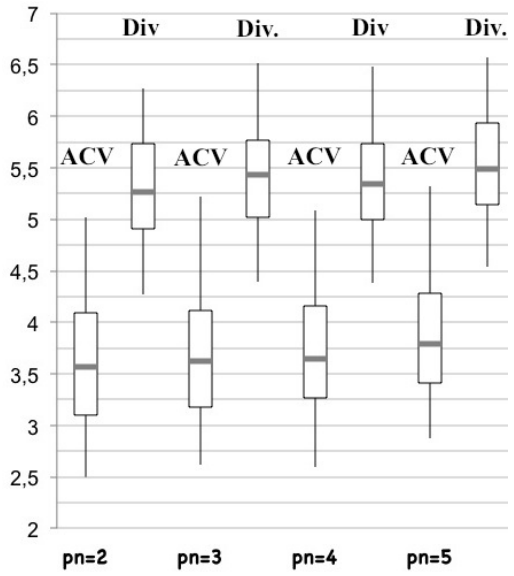

Fig. 6.117: Performance of multi-parent ExpoDeRP with $ps = 100$, (3D-MOP)

ge from 2 to 5. The diversity is nearly comparable for $pn = 2$ to $pn = 4$, the highest diversity is achieved for $pn = 5$. The convergence reveals a tendency towards increased ACV values and therefore to a slowed down convergence by an increase of pn. The performance of the configurations with decreased ps values of 60 and 80 are depicted in Figure 6.115 and 6.116. Generally, there is no regularity observable by an increase of pn, regarding convergence and diversity. The diversity is oscillating for $ps = 60$ (Figure 6.115): The diversity for $pn = 2$ and $pn = 4$ are comparable and remarkably higher than the diversity

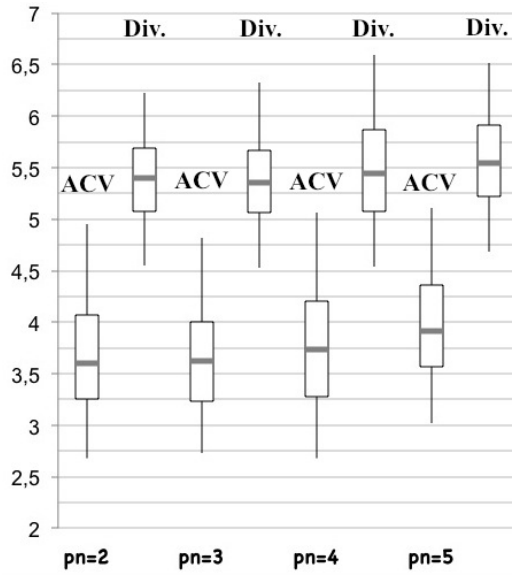

Fig. 6.118: Performance of multi-parent ExpoDeRP with $ps = 120$, (3D-MOP)

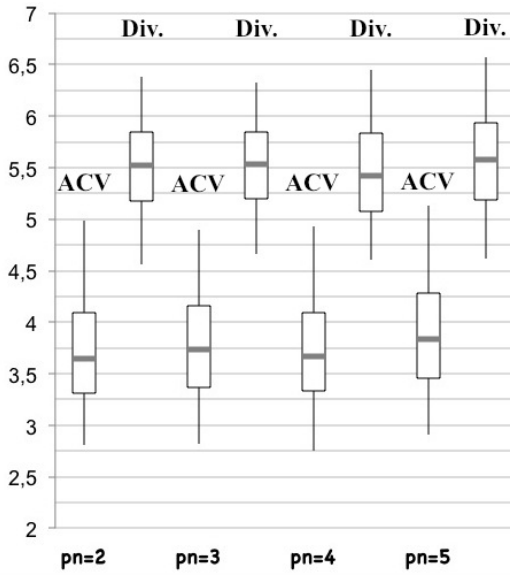

Fig. 6.119: Performance of multi-parent ExpoDeRP with $ps = 140$, (3D-MOP)

values of $pn = 3$ and $pn = 5$. The convergence results of $pn = 2, 3$ and $pn = 5$ are comparable, the results of $pn = 4$ are remarkably increased and therefore reveal a lower convergence. The convergence results of the configurations with $ps = 80$ are oscillating, the ACV values of $pn = 2$ and $pn = 4$ are higher than those of $pn = 3$ and $pn = 5$. The diversity values for $pn = 2$ and $pn = 5$ are higher than those of $pn = 2$ and $pn = 3$. Figure 6.118 and 6.119 present the performance results of the configurations with $ps = 120$ and 140. In the case of $ps = 120$, the convergence as well as diversity performance are comparable for $pn = 2$ and $pn = 4$. The diversity and convergence values are increased for higher pn values, revealing an improvement of the diversity at the cost of convergence. In the case of $ps = 140$, convergence and diversity are nearly comparable for $pn = 2$ to $pn = 4$. The diversity and convergence values are increased for $pn = 5$.

The performance results of a parent number variation achieved by configurations with different population sizes in the case of the 4D-MOP are depicted in the figures 6.120 to 6.124. Figure 6.122 presents the performance results of the configurations with $ps = 100$. The increase of pn results in an increase of diversity as well as the ACV values until $pn = 4$. The performance results of $pn = 5$ are on the same level as those of $pn = 3$. Figure 6.120 depicts the performance results of the configurations with $ps = 60$. Convergence and diversity reveal a slight oscillating behavior: The ACV and diversity values tend to be lower for $pn = 2$ and $pn = 4$ compared to the other pn settings. In the case of the configurations with $ps = 80$ (Figure 6.121), there is a remarkable

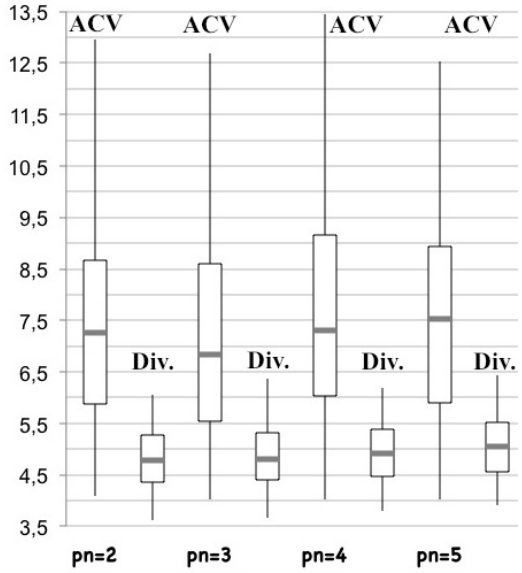

Fig. 6.120: Performance of multi-parent ExpoDeRP with $ps = 60$, (4D-MOP)

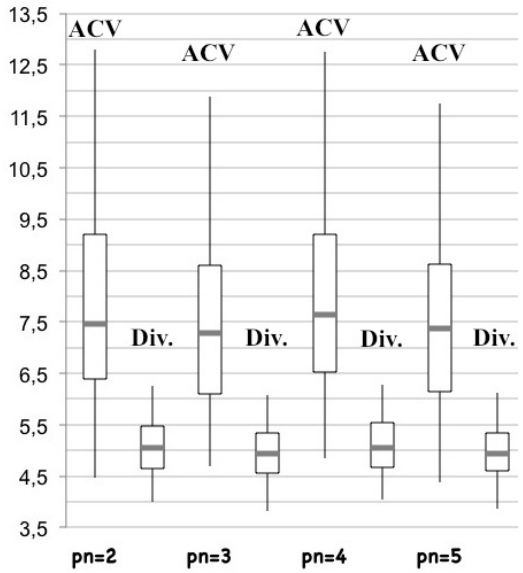

Fig. 6.121: Performance of multi-parent ExpoDeRP with $ps = 80$, (4D-MOP)

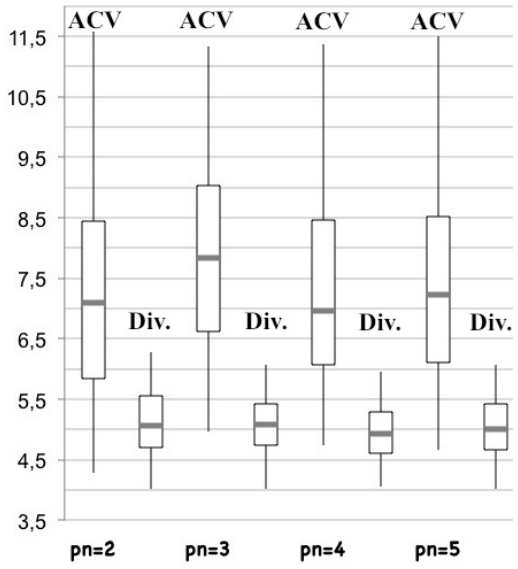

Fig. 6.122: Performance of multi-parent ExpoDeRP with $ps = 100$, (4D-MOP)

increase of diversity and a slight increase of the ACV values observable by an increase of $pn = 2$ to $pn = 3$. A further increase of pn settings results in similar performance results of convergence and diversity. In the case of the configurations with $ps = 120$ (Figure 6.123), the increase of pn results in an increase of the ACV values and therefore in lower convergence. Otherwise, the diversity is significantly reduced by an increase of $pn = 2$ to $pn = 3$. A further increase of pn reveals a remarkable increase of the diversity values. Figure 6.124 depicts the performance results of the configurations with $ps = 140$. An increase of

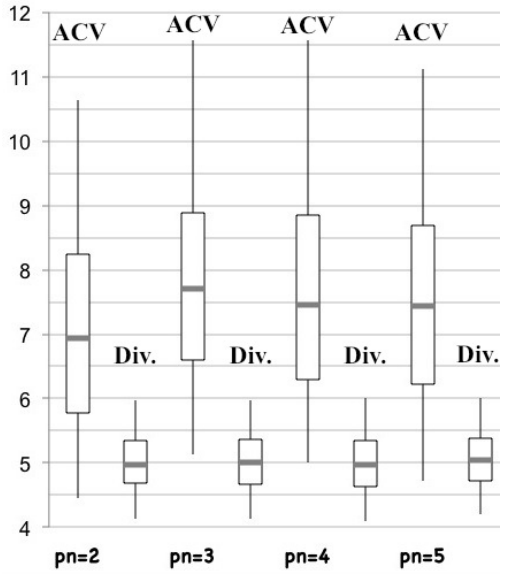

Fig. 6.123: Performance of multi-parent ExpoDeRP with $ps = 120$, (4D-MOP)

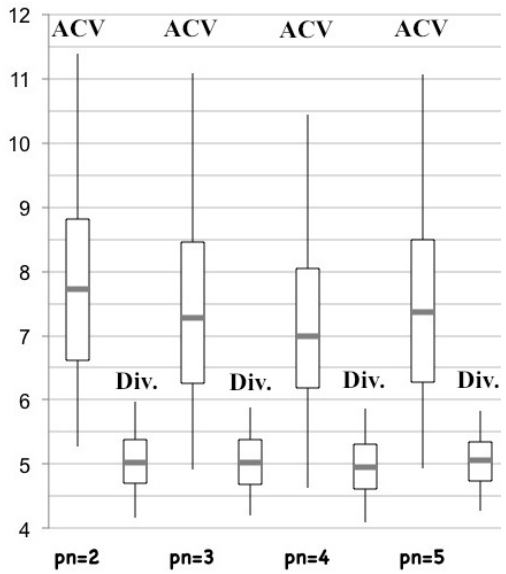

Fig. 6.124: Performance of multi-parent ExpoDeRP with $ps = 140$, (4D-MOP)

pn results in a slight oscillating performance and the ACV values are slightly increased, revealing lower convergence values.
The comparison of the performance results in the case of the 3D- and 4D-MOP reveals that there is no interdependence observable with regard to a variation of pn for different ps settings. Interdependent of the population size, pn settings higher than 3 result in increased diversity at the cost of convergence in most cases. The performance results of the 3D- and 4D-MOP provide no regularity regarding convergence and diversity by an increase of pn for ps smaller than 100, whereas the 3D-performance results for $ps = 100$ and higher reveal increased performance results for pn higher than 3, indicating improved diversity at the cost of convergence. The 3D-performance of the configurations with $pn = 2$ and $pn = 3$ are comparable in most cases.

6.3.3 Discussion

The influence of multi-parent recombination is investigated in the case of the 3D- and 4D-MOP with the recombination operators LiDeRP and ExpoDeRP. The investigations of parent number variation in the case of LiDeRP and ExpoDeRP reveal different performance results. Both recombination operators with a variation of the parent number and population size reveal comparable performance results for both MOPs.
Result matrices are presented for the parent number variation of LiDeRP to give an overview of the multi-parent number influence on the performance and

3D	ps=60	ps=80	ps=100	ps=120	ps=140
pn=2	[3.23, 3.99] [5.07, 5.78]	[3.52, 4.37] [5.23, 5.87]	[3.22, 3,97] [5.02, 5.57]	[3.39, 3.95] [5.16, 5.62]	[3.46, 3.99] [5.26, 5.75]
pn=3	[3.22, 3.99] [4.95, 5.64]	[3.43, 4.13] [5.12, 5.68]	[3.37, 4.00] [5.16, 5.69]	[3.38, 3.91] [5.18, 5.59]	[3.50, 4.04] [5.32, 5.73]
pn=4	[3.37, 4.23] [5.11, 5.77]	[3.45, 4.37] [5.14, 5.74]	[3.37, 4.05] [5.11, 5.64]	[3.43, 4.08] [5.22, 5.75]	[3.45, 3.98] [5.23, 5.70]
pn=5	[3.18, 3.94] [4.95, 5.60]	[3.42, 4.12] [5.25, 5.85]	[3.51, 4.25] [5.26, 5.79]	[3.66, 4.27] [5.32, 5.80]	[3.61, 4.14] [5.35, 5.83]

Table 6.7: Result matrix of the 3D-MOP presenting the performance of the *ps* and *pn* variation with LiDeRP. The upper intervals in each cell represent the confidence limits of the convergence means, the lower intervals represent the confidence limits of the diversity means. The optimal performance results are highlighted in green, the worst in red and the blue one highlights the compromise results

4D	ps=60	ps=80	ps=100	ps=120	ps=140
pn=2	[6.48, 8.52] [4.53, 5.12]	[6.86, 8.85] [4.77, 5.36]	[6.42, 8.12] [4.84, 5.40]	[6.30, 7.91] [4.77, 5.26]	[7.09, 8.67] [4.81, 5.31]
pn=3	[6.18, 8.32] [4.57, 5.21]	[6.63, 8.40] [4.69, 5.23]	[7.11, 8.68] [4.83, 5.33]	[7.06, 8.61] [4.79, 5.26]	[6.72, 8.18] [4.81, 5.27]
pn=4	[6.63, 8.91] [4.61, 5.28]	[7.02, 8.94] [4.83, 5.40]	[6.61, 8.13] [4.73, 5.21]	[6.93, 8.50] [4.77, 5.25]	[6.50, 7.83] [4.75, 5.20]
pn=5	[6.58, 8.62] [4.74, 5.41]	[6.61, 8.38] [4.69, 5.25]	[6.70, 8.18] [4.77, 5.29]	[6,70, 8.35] [4.84, 5.30]	[6.73, 8.26] [4.86, 5.24]

Table 6.8: Result matrix of the 4D-MOP presenting the performance of the *ps* and *pn* variation with LiDeRP. The upper intervals in each cell represent the confidence limits of the convergence means, the lower intervals represent the confidence limits of the diversity means. The optimal performance results are highlighted in green, the worst in red and the blue one highlights the compromise results.

of the interdependence between parent number variation and population size in the case of the 3D-MOP (Table 6.7) and the 4D-MOP (Table 6.8). The upper interval in each cell is the confidence limit of ACV means, and the lower interval is the confidence interval of diversity means. Best and worst performance of the configurations with different population sizes are highlighted. Green highlighted cells mark the best performance results in the sense of a comparable good convergence-diversity balance. Red highlighted cells mark the worst performance results in the sense of the highest metric values or an inadequate convergence-diversity balance. A blue box highlights the configuration which represents a compromise between the second highest ACV confidence limits and comparable diversity values. The results matrices do not reveal a specific interaction between the population size and the parent number for recombination. Furthermore, there is no particular parent number that achieves good

performance results independent of the population size. The 3D-result matrix (Table 6.7) reveals that an increase of the parent number from 2 to 3 results in increased or comparable diversity results with mostly comparable convergence results. An increase of the parent number from 3 to 4 reveals increased diversity for $ps \geq 100$ at the cost of convergence and decreased diversity for $ps < 100$. A further increase of the parent number from 4 to 5 reveals a considerable diversity increase at the cost of convergence, especially for $ps \geq 100$. As a consequence, the parent number increase from 2 to 3 is the most successful with regard to diversity increase with comparable convergence results at the same time.

The 4D-result matrix (Table 6.8) reveals similar performance results compared to the 3D-results matrix: The parent number increase from 2 to 3 mostly results in comparable diversity results with improved convergence. A further increase of the parent number from 3 to 4 results in decreased diversity in the case of $ps \geq 100$ and improved convergence. A diversity increase is achieved by an increase from 4 to 5 parents, mostly at the cost of convergence. Once again, the increase from 2 to 3 parents is most successful.

The experimental results of the multi-parent variation with ExpoDeRP reveal no regularity regarding convergence and diversity performance. In general, there is no general interdependence observable between parent number and population size. The 3D-result matrix (Table 6.9) reveals a diversity increase at the cost of convergence in the case of $ps \geq 100$ and an increase of 2 to 3 parents for recombination. The increase of the parent number from 3 to 4 mostly reveals a diversity increase for $ps \geq 100$ and a diversity decrease for $ps < 100$. The convergence performance is indifferentiable at the same time. A further increase of the parent number from 4 to 5 achieved increased diversity at the cost of convergence.

The 4D-result matrix (Table 6.10) reveals that an increase of the parent number mostly results in a diversity increase at the cost of convergence. The issue of an advisable parent number for ExpoDeRP is rather complex, since the diversity improvement is mostly at the cost of convergence. Generally, a parent number lower than five is advisable, since the worst performance results are achieved for $pn = 5$ independent of the population size. Furthermore, the 3D-result matrix allow the conclusion that a lower parent number is more advisable in combination with a population size of 100 and higher in the case of the 3D-MOP.

Summarizing, advisable settings for the parent number depend on the recombination type: A higher parent number in combination with a population size of 100 and higher is more advisable in the case of LiDeRP, a parent number

3D	ps=60	ps=80	ps=100	ps=120	ps=140
pn=2	[3.30, 4.17] [5.08, 5.79]	[3.48, 4.20] [5.22, 5.91]	[3.32, 3,93] [5.12, 5.63]	[3.31, 3.87] [5.15, 5.60]	[3.45, 4.04] [5.21, 5.70]
pn=3	[3.38, 4.19] [5.10, 5.72]	[3.44, 4.14] [5.18, 5.76]	[3.53, 4.13] [5.23, 5.73]	[3.46, 3.98] [5.21, 5.68]	[3.50, 4.00] [5.31, 5.78]
pn=4	[3.30, 4.10] [5.20, 5.80]	[3.55, 4.22] [5.24, 5.82]	[3.40, 3.96] [5.11, 5.59]	[3.62, 4.23] [5.35, 5.85]	[3.48, 4.04] [5.28, 5.71]
pn=5	[3.53, 4.26] [5.07, 5.73]	[3.53, 4.25] [5.21, 5.77]	[3.43, 4.01] [5.25, 5.74]	[3.53, 4.08] [5.33, 5.78]	[3.62, 4.20] [5.39, 5.83]

Table 6.9: Result matrix of the 3D-MOP presenting the performance of the *ps* and *pn* variation with ExpoDeRP. The upper intervals in each cell represent the confidence limits of the convergence means, the lower intervals represent the confidence limits of the diversity means. The optimal performance results are highlighted in green, the worst in red and the blue one highlights the compromise results.

4D	ps=60	ps=80	ps=100	ps=120	ps=140
pn=2	[6.54, 8.80] [4.73, 5.47]	[6.66, 8.54] [4.69, 5.26]	[6.62, 8.35] [4.72, 5.25]	[7.00, 8.57] [4.74, 5.18]	[7.11, 8.65] [4.79, 5.26]
pn=3	[7.08, 9.21] [4.85, 5.42]	[6.98, 9.00] [4.81, 5.44]	[7.04, 8.83] [4.86, 5.37]	[6.72, 8.16] [4.80, 5.27]	[7.24, 8.66] [4.88, 5.30]
pn=4	[6.93, 8.87] [4.69, 5.29]	[7.02, 8.98] [4.91, 5.47]	[7.20, 9.00] [4.99, 5.53]	[7.00, 8.45] [4.95, 5.37]	[7.01, 8.49] [4.92, 5.34]
pn=5	[6.97, 9.02] [4.83, 5.44]	[7.11, 9.04] [4.87, 5.41]	[7.27, 8.95] [4.88, 5.36]	[7.30, 8.79] [4.98, 5.42]	[7.22, 8.76] [5.02, 5.46]

Table 6.10: Result matrix of the 4D-MOP presenting the performance of the *ps* and *pn* variation with ExpoDeRP. The upper intervals in each cell represent the confidence limits of the convergence means, the lower intervals represent the confidence limits of the diversity means. The optimal performance results are highlighted in green, the worst in red and the blue one highlights the compromise results.

smaller than five with a population size of 100 and higher is generally more advisable in the case of ExpoDeRP.

The following theoretical considerations emphasize the experimental results: From a theoretical point of view, a lower parent number for recombination produces peptides which are very similar to their parents according to their primary structure, whereas a high parent number produces peptides which are very different in their primary structure compared to their parent. On the basis of these considerations, $pn = 3$ is regarded as an adequate choice for the parent number in recombination: The configurations with $pn = 5$ mainly result in worse performance, the offspring of the configurations with $pn = 4$ differ more in the genetic appearance compared to their parents than in the case of $pn = 3$ and $pn = 2$ and therefore causes uncontrollable fitness jumps. The setting $pn = 2$ does not comply with the aspired genetic diversity concept. This makes a setting of $pn = 3$ highly recommended.

6.4 Selection Pressure

The experiments presented in this section provide an insight into the impact of the interdependence between selection procedure and population size. The impact is examined with the objective of giving a configuration rule for the population size and the selection parameter p_0, which has shown a significant influence on the VONSEA performance. Furthermore, these experiments are an empirical verification of the focused early convergence behavior of VONSEA within a limited range of population sizes. Otherwise, the early convergence is defeated, since an unlimited increase of the population size results in improved convergence and diversity performance. More precisely, this section answers the questions raised in the research issue section of this thesis: Firstly, do larger populations speed up the convergence behavior? Secondly, is there a predictable impact between population size and selection procedure? An import request is the justification of a configuration rule for the range of population sizes and suitable selection parameters ensuring a good VONSEA performance, exemplarily examined in the case of the 3D- and 4D-MOP.

6.4.1 Experiments with ACV-based Selection

The interdependence is examined between the population size and the three selection strategies. The other components of the investigated configurations are BS_{adapt} mutation, LiDeRP as recombination, three parents for recombination and $ts = 10$. The selection of $ts = 10$ refers to the observations and consequences of the experimental results presented in Section 6.2.3. The motivation for the recombination parameter pn is given in Section 6.3. These components have proven to provide a good VONSEA performance on the 3D- and 4D-MOP. The configurations are further composed of a different population size (30, 60, 80, 90, 100, 110, 120, 140) and a variation of p_0, depending on the selection strategy. The test runs are once again evaluated by the ACV indicator that has especially evolved to compare the convergence performance of differently sized populations: ACV_{scaled} is used as convergence indicator to ensure an insight into the convergence performance with an equal influence of each objective function on the indicator value. The ACV_{rel} results are further proposed to reflect the relative non-dominated solution quality. The proposed diversity indicator also takes account of the population size and therefore allows a comparison of different sized populations.

Figure 6.125 presents the 3D-performance results of the configurations with

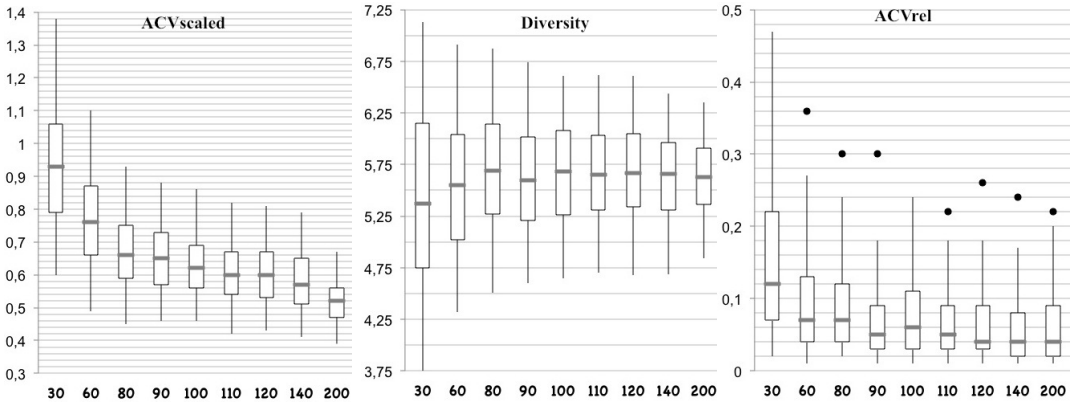

Fig. 6.125: 3D-Performance results of configurations with ACV-based selection ($p_0 = 40\%$) and a variation of ps.

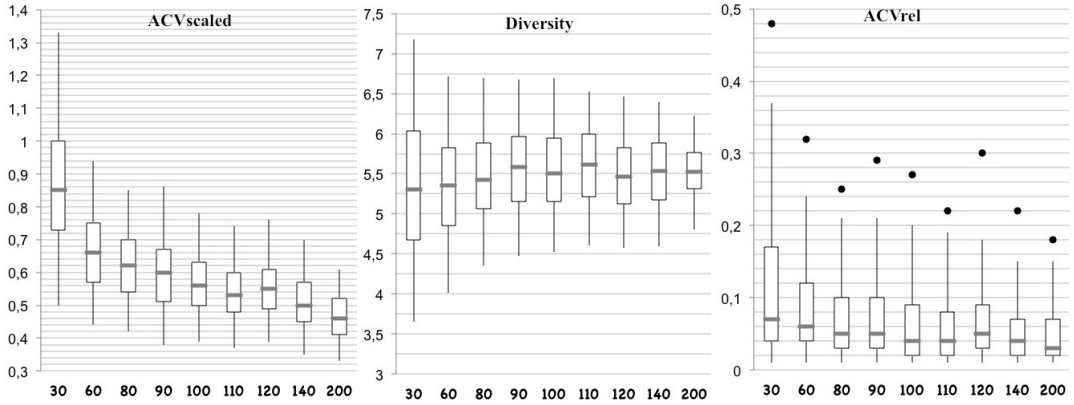

Fig. 6.126: 3D-Performance results of configurations with ACV-based selection ($p_0 = 50\%$) and a variation of ps.

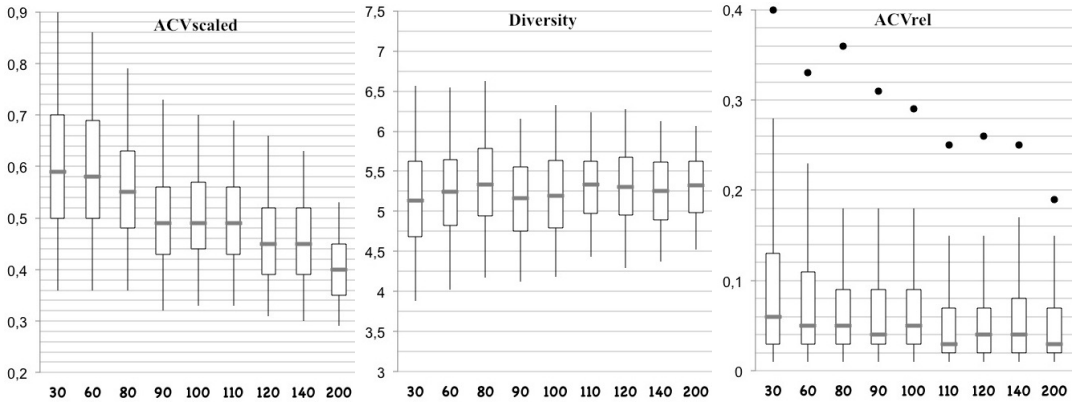

Fig. 6.127: 3D-Performance results of configurations with ACV-based selection ($p_0 = 60\%$) and a variation of ps.

ACV-based selection, the selection parameter $p_0 = 40\%$ and a variation of ps ($ps = 30, 60, 80, 90, 100, 110, 120, 140, 200$). The left figure presents the ACV_{scaled} values, the right figure depicts the ACV_{rel} results and diversity is presented in the middle figure. The increase of ps results in a significant de-

crease of the ACV_{scaled} values as well as a decrease of the range of ACV_{scaled} values and therefore in a continuous improvement of the convergence behavior. The level of decrease is remarkably higher in the case of ps under 100. The level of decrease flattens for ps above 100. The ACV_{scaled} results are comparable for $ps = 110$ and 120. The diversity performance reveals a tendency towards higher diversity values for ps settings of 80 and higher. The range of diversity values is also decreased by an increase of ps. The relative quality of the non-dominated solutions is generally decreased for increased ps values. Therefore, the convergence quality of the non-dominated solutions is relative high compared to the convergence quality of the entire population. There is no significant ACV_{rel} improvement for ps values higher than 110. In general, the decrease of the ACV_{rel} values is observable by an increase of ps of 30 up to 90. The performance results of the different configurations with varying ps values reveals skewed results as well as outliers for nearly every configuration indicating that ACV_{rel} results have remarkably higher indicator values.
The performance results of the configurations with ACV-based selection, $p_0 = 50\%$ - the advisable selection parameter setting according to Section 6.2 - and a variation of ps are depicted in Figure 6.126. The ACV_{scaled} values in general as well as the range of ACV_{scaled} values are decreased significantly by an increase of ps as in the case of the configurations with $p_0 = 40\%$. The ACV_{scaled} results are nearly comparable for $ps = 110$ and $ps = 120$. The diversity results reveal that an increase of the diversity is observable by an increase of $ps = 60$ up to $ps = 110$. For higher ps values, the diversity values are generally reduced compared to the other configurations. The relative quality of the non-dominated solutions is decreased by an increase of $ps = 30$ to $ps = 110$. The ACV_{rel} results are comparable for $ps = 80$ and $ps = 90$. Furthermore, the ACV_{rel} values are significantly decreased for ps higher than 120. Once, again, the ACV_{rel} values are skewed revealing a tendency to remarkably higher indicator values. In general, the three metric values are decreased compared to the configurations with $p_0 = 40\%$ (Figure 6.125), indicating improved convergence and non-dominated solutions quality at the cost of diversity. This effect has also been reported in the chapter 6.2.
The performance results of the configurations with $p_0 = 60\%$ and a variation of ps is depicted in Figure 6.127. The ACV_{scaled} performance is nearly comparable for $ps = 30$ and $ps = 60$ as well as for the configurations with $ps = 90$ to $ps = 110$ and generally decreased but similar for $ps = 120$ and 140. Generally, ACV_{scaled} values decrease by an increase of ps. The diversity performance increases by an increase of $ps = 30$ up to $ps = 80$. The diversity values are generally lower but similar in the cases of ps settings higher than $ps = 80$. Also

the ACV_{rel} values decrease by an increase of ps. The ACV_{rel} values are comparable for $ps = 80$ to $ps = 100$. Once again, the ACV_{rel} values are skewed for all configurations with different ps settings. Furthermore, the metric values are once again decreased compared to the configurations with $p_0 = 40\%$ (Figure 6.125) and $p_0 = 50\%$ (Figure 6.126). The increase of the selection parameter p_0 reduces the decrease of the ACV_{scaled} as well as the ACV_{rel} values by an increase of ps.

The decrease of the ACV_{scaled} and ACV_{rel} values is observable for all ps settings by an increase of ps. Furthermore, no improvement of the diversity values is observable by an increase of ps above 110. As a consequence, an advisable range of ps settings regarding the performance results is in the range of 80 to 100. The slight performance improvement for ps values higher than 110 regarding the convergence and non-dominated solution quality is in contrast to the higher computational complexity.

The performance results of the configurations with the ACV-based selection

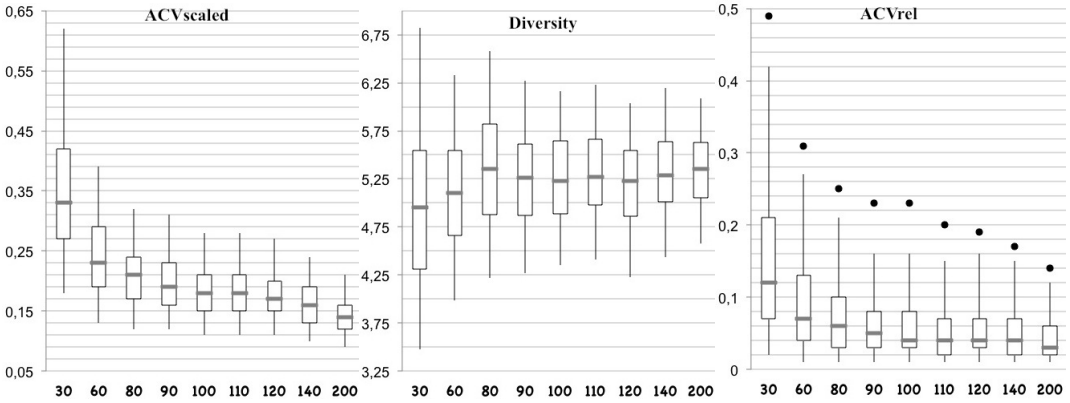

Fig. 6.128: 4D-Performance results of configurations with ACV-based selection ($p_0 = 40\%$) and a variation of ps.

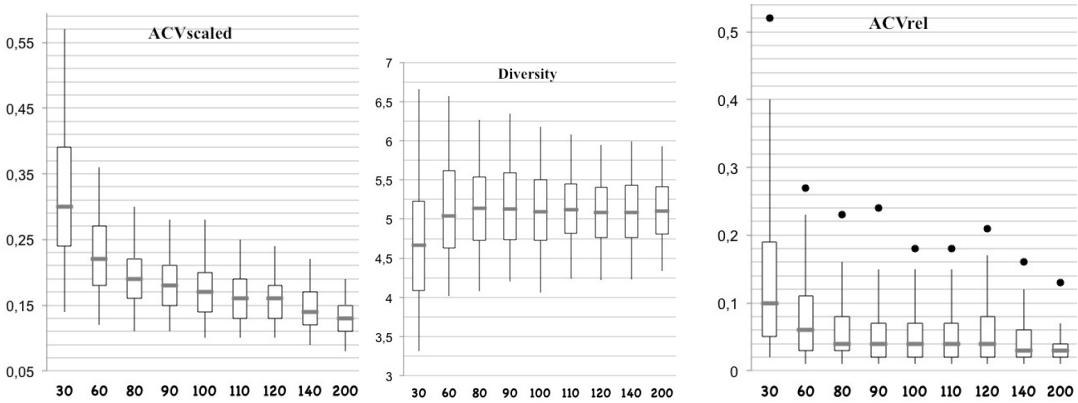

Fig. 6.129: 4D-Performance results of configurations with ACV-based selection ($p_0 = 50\%$) and a variation of ps.

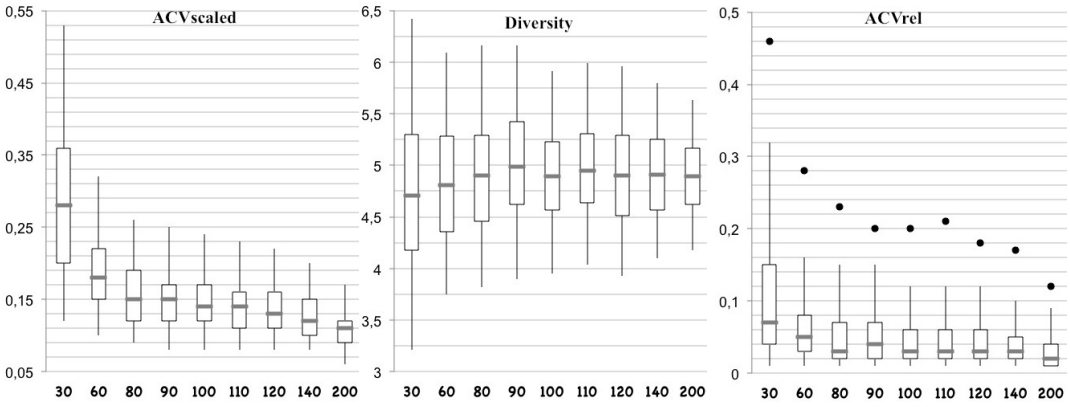

Fig. 6.130: 4D-Performance results of configurations with ACV-based selection ($p_0 = 60\%$) and a variation of ps.

and a variation of ps in the case of the 4D-MOP are presented in the following. The results are discussed in comparison to the corresponding 3D-performance results. Figure 6.128 depicts the 4D-performance results of the configurations with the selection parameter $p_0 = 40\%$. The ACV_{scaled} values are in general significantly decreased as well as the range of ACV_{scaled} values by an increase of ps, as in the case of the corresponding 3D-performance (Figure 6.125). Furthermore, the level of decrease is reduced for ps values higher than 110. Moreover, the ACV_{scaled} performance of $ps = 10$ and $ps = 110$ are comparable. The diversity performance is similar for $ps = 90$ and higher. The diversity performance provides a wide range of indicator values for $ps = 30$ and the highest diversity performance is achieved for $ps = 80$. The ACV_{rel} values are decreased by an increase of ps, as in the case of the corresponding 3D-performance results (Figure 6.125). Once again, the level of decrease is reduced for ps values higher than 100. However, the ACV_{rel} results are skewed, which is visible by the outliers and reveals a tendency to remarkably higher performance results. The 4D-performance results of the configurations with the selection parameter $p_0 = 50\%$ and a variation of ps are presented in Figure 6.129. The ACV_{scaled} results reveal the same observations as in the case of the configurations with $p_0 = 40\%$ and the corresponding 3D results (Figure 6.126): The ACV_{scaled} values are decreased by an increase of ps and the level of decrease is reduced for higher ps values. The diversity performance provides a wide range of indicator values for $ps = 30$ and becomes more comparable for $ps = 60$ and higher. The diversity values are slightly reduced and comparable by an increase of ps above $ps = 100$. Once again, a decrease of the ACV_{rel} values is observable by an increase of ps. The ACV_{rel} results are comparable for $ps = 90$ to $ps = 110$. Moreover, the ACV_{rel} results are skewed, which is also visible by the number of outliers. Furthermore, this relative non-dominated solution quality is also

comparable to the corresponding 3D-performance results (Figure 6.126).
The 4D-performance of the configurations with $p_0 = 60\%$ reveals once again a significant decrease of the ACV_{scaled} and therefore a convergence improvement by an increase of ps. The ACV_{scaled} results are comparable for $ps = 90$ and 100 as well as for the configurations with $ps = 110$ and 120, though slightly decreased. The diversity performance reveals only slight diversity variations and are nearly comparable for the configurations with $ps = 100$ and higher ps values. The highest diversity performance is achieved for $ps = 90$. Once again, the ACV_{rel} values are remarkably decreased by an increase of ps, as in the case of the configurations with $p_0 = 40\%$ and 50% and the corresponding 3D-performance results (Figure 6.130).
Summarizing, the 4D-performance results of the configurations with ACV-based selection, different selection parameter settings for p_0 and a variation of ps reveal comparable ACV_{scaled} and ACV_{rel} by an increase of ps and for each p_0: An increase of ps reveals a continuous decrease of the ACV_{scaled} and ACV_{rel} values independent of the p_0 settings. Furthermore, these results are comparable to the corresponding 3D results. At the same time, the diversity results do not reveal any significant improvement by an increase of ps. Moreover, the diversity results are becoming more and more comparable for higher ps settings and increasing selection parameters p_0. Otherwise, an increase of the selection parameter p_0 results in a general decrease of the metric values, as in the case of the 3D-performance results. An advisable range of ps settings regarding a performance improvement and an appropriate computational complexity is given by a range of $ps = 90$ to $ps = 110$. The performance improvement caused by higher ps values forms a contrast to the remarkably higher computational complexity.

6.4.2 Experiments with ACV-random Selection

The following experimental results demonstrate the convergence, diversity and relative non-dominated solution quality of the configurations with ACV-random selection, different selection parameter settings of p_0 and a variation of ps in the case of the 3D-MOP. These results are also discussed in contrast to the performance of the configurations with ACV-based selection presented above. Figure 6.131 presents the 3D-performance results of the configurations with $p_0 = 0\%$, indicating a pure front-based SUS. The increase of ps reveals a continuous decrease of the convergence performance, visible by the decrease of the ACV_{scaled} values, whereas the decrease is reduced by ps values higher

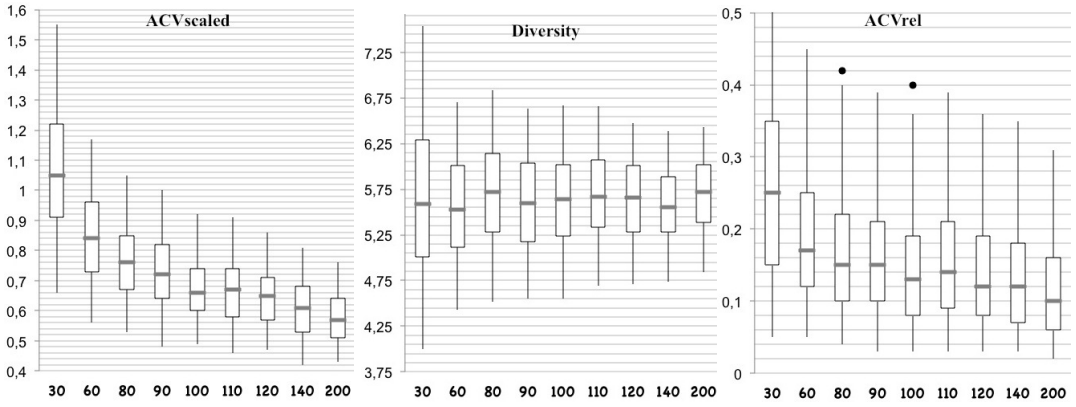

Fig. 6.131: 3D-Performance results of configurations with ACV-random selection ($p_0 = 0\%$) and a variation of ps.

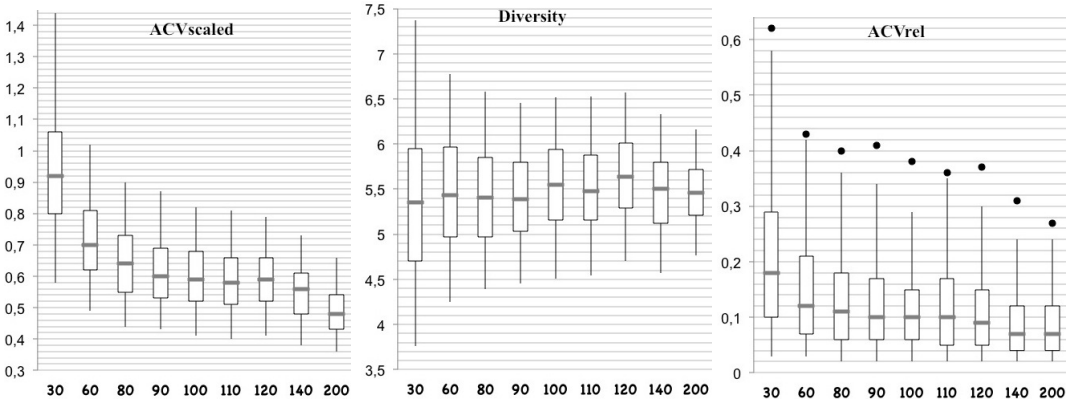

Fig. 6.132: 3D-Performance results of configurations with ACV-random selection ($p_0 = 10\%$) and a variation of ps.

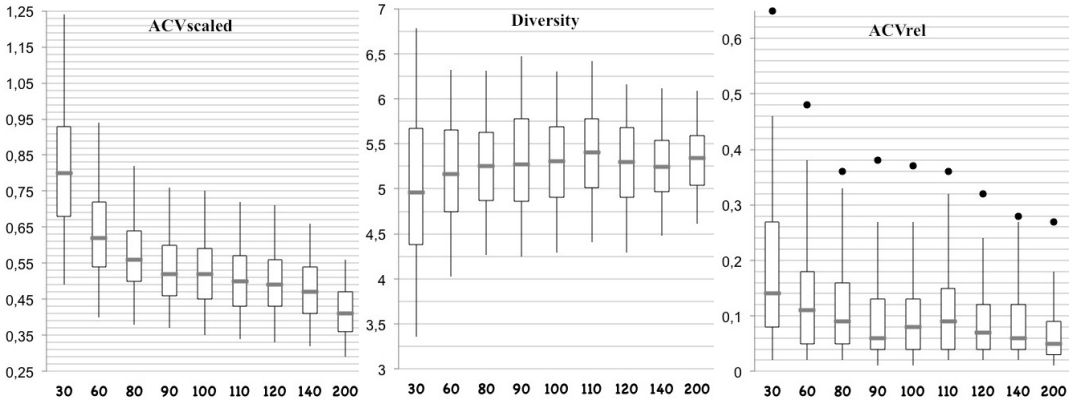

Fig. 6.133: 3D-Performance results of configurations with ACV-random selection ($p_0 = 20\%$) and a variation of ps.

than 100. This convergence behavior is comparable to the general convergence performance of the configurations with ACV-based selection. The ACV_{scaled} results of $ps = 100$ and 110 are comparable. The diversity behavior reveals generally no regularity by an increase of ps. However, the diversity perfor-

mance is generally reduced by an increase of ps. The highest diversity results are achieved for $ps = 30$ and $ps = 80$, the range of the diversity results is reduced by increasing ps settings. The relative non-dominated solution quality is improved by an increase of ps, visible by the ACV_{rel} values. The ACV_{rel} results are comparable for the configurations with $ps = 100$ and $ps = 120$. The ACV_{rel} results are also comparable to the general convergence performance of configurations with ACV-based selection.

Figure 6.132 presents the 3D-performance results of the configurations with $p_0 = 10\%$ and the advisable selection parameter setting according to Section 6.2. Once again, the convergence is improved by an increase of ps, the convergence behavior of the settings $ps = 100$ and $ps = 120$ are nearly comparable. The diversity performance reveals no regularity by an increase of ps. The performance differences between the configurations with different ps settings are reduced. The diversity performance is generally reduced for ps higher than 140. The ACV_{rel} values are once again decreased by an increase of ps. The ACV_{rel} results of $ps = 100$ and 120 as well as $ps = 140$ and 200 are comparable. The ACV_{rel} performance differences are reduced compared to the configurations with $p = 0\%$. The general performance tendency of ACV_{scaled} and ACV_{rel} values are comparable to the general performance tendency of the ACV-based configurations.

These observations are transferable to the performance of the configurations with $ps = 20\%$ (Figure 6.133). The ACV_{scaled} values are decreased by an increase of ps as well as the ACV_{rel} values. The highest diversity results are achieved for $ps = 90$ to $ps = 110$. The range of the diversity results are reduced as well as the diversity values in general for $ps = 140$ and 200.

In the following, the 4D-performance results of the configurations with ACV-

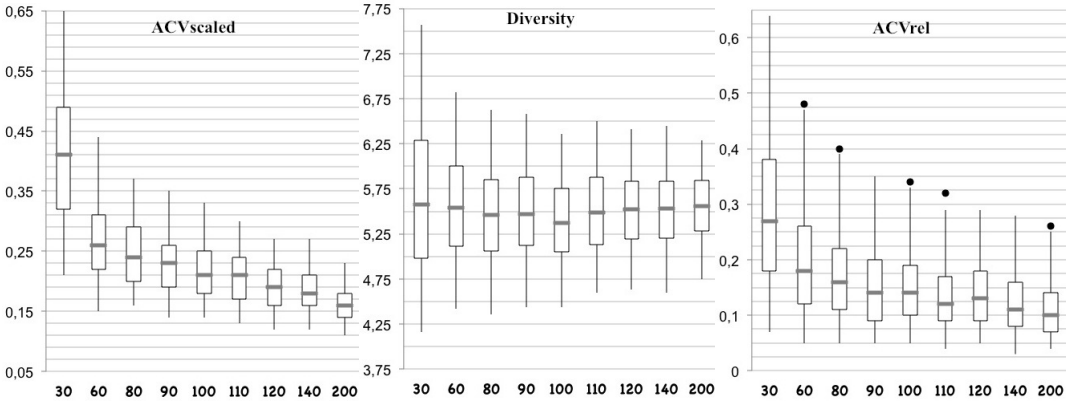

Fig. 6.134: 4D-Performance results of configurations with ACV-random selection ($p_0 = 0\%$) and a variation of ps.

random selection and a variation of ps are presented. The experimental results

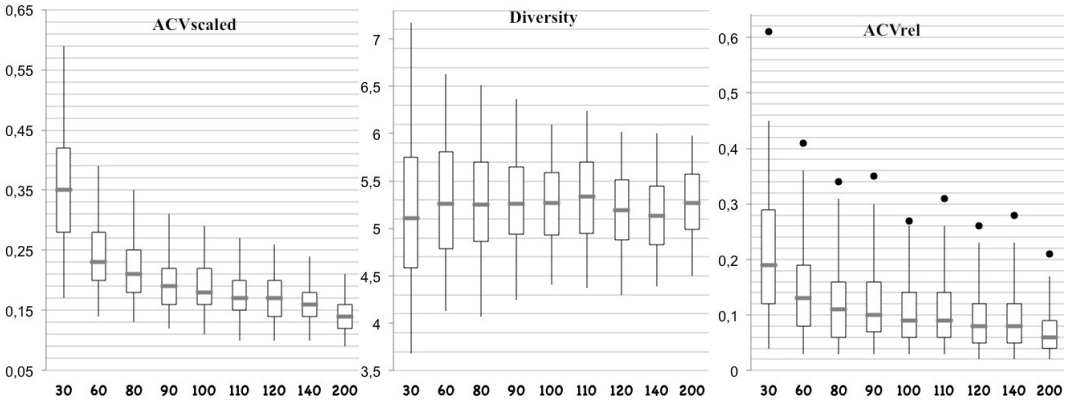

Fig. 6.135: 4D-Performance results of configurations with ACV-random selection ($p_0 = 10\%$) and a variation of ps.

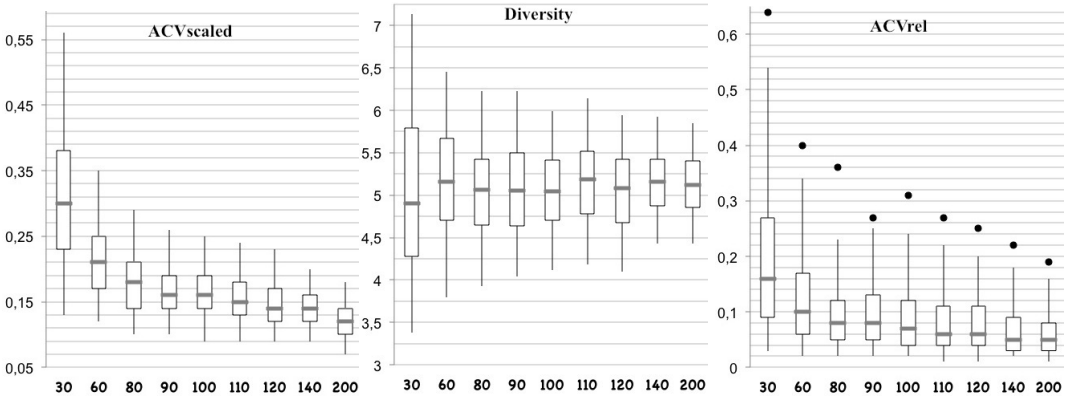

Fig. 6.136: 4D-Performance results of configurations with ACV-random selection ($p_0 = 20\%$) and a variation of ps.

evaluated in the following present the 4D-performance results of the configurations with ACV-random selection, different p_0 values and a variation of ps. The convergence and the relative non-dominated solution quality of the configurations with $p_0 = 0\%$ - indicating front-based SUS selection - are improved by an increase of ps, visible by the decreasing ACV_{scaled} and ACV_{rel} results (Figure 6.134). These performance results have already been achieved by the corresponding 3D-performance results. The diversity performance reveals a significant decrease in the range of $ps = 30$ to $ps = 100$. Diversity is slightly increased for ps values higher than 100.

The ACV_{scaled} and ACV_{rel} results of the configurations with $p_0 = 10\%$ are already familiar to the previous 3D-experimental results with regard to the improvement by an increase of ps (Figure 6.135). Furthermore, the ACV_{rel} results of the configurations with $ps = 100$ and $ps = 140$ are comparable as well as $ps = 120$ and $ps = 140$, though reduced. Moreover, the performance improvement is reduced for higher ps values. The diversity results are decrea-

sed in general as well as for ps values higher than $ps = 110$.
The decrease of the ACV_{scaled} and ACV_{rel} values by an increase of ps is also observable in the case of the configurations with $p_0 = 20\%$, whereas the level of decrease is reduced compared to the previous experimental results, especially in the case of the ACV_{rel} values (Figure 6.136). The ACV_{rel} values are nearly comparable for $ps = 80$ to $ps = 120$ and slightly reduced for higher ps values. The range of diversity values is reduced for $ps = 80$ and higher ps values.

Summarizing, these 3D- and 4D-performance results of the configurations with ACV-random selection reveal that these results are generally comparable to the experimental results of the configurations with ACV-based selection regarding the influence of ps and p_0 on the performance: The convergence as well as the relative non-dominated solution quality is improved by an increase of ps for every p_0 setting. Furthermore, the performance differences between the different ps settings are reduced by an increase of the p_0 settings. The diversity performance does not reveal any general regularity depending on the p_0 or ps variation. Otherwise, the diversity performance becomes more comparable for the different ps settings in the case of higher p_0 values. Beneath the relative non-dominated performance improvement by an increase of ps, the ACV_{rel} values become more comparable for higher p_s values in combination with higher p_0 settings. Furthermore, these results are usually skewed revealing a tendency towards remarkably higher indicator values. In general, the metric values are reduced by an increase of p_0. This is also observable in the case of the configurations with the ACV-based selection presented above and in the test series of Section 6.2. As the indicator results of ACV_{rel} and diversity are nearly comparable for ps values in a range of 80 to 120 and the performance improvement of the ACV_{scaled} values is relative small compared to the increase of the computational complexity caused by higher ps values, an advisable range of ps settings is the interval of $ps = 90$ to $ps = 110$.

6.4.3 Experiments with Aggregate Selection

The following series of tests present the 3D-performance results of the configurations with aggregate selection, different p_0 settings and a variation of ps. The convergence behavior of the configurations with $p_0 = 50\%$ (Figure 6.137) is continuously improved by an increase of ps, such as in the other series of tests presented above. The range as well as the tendency towards maximal

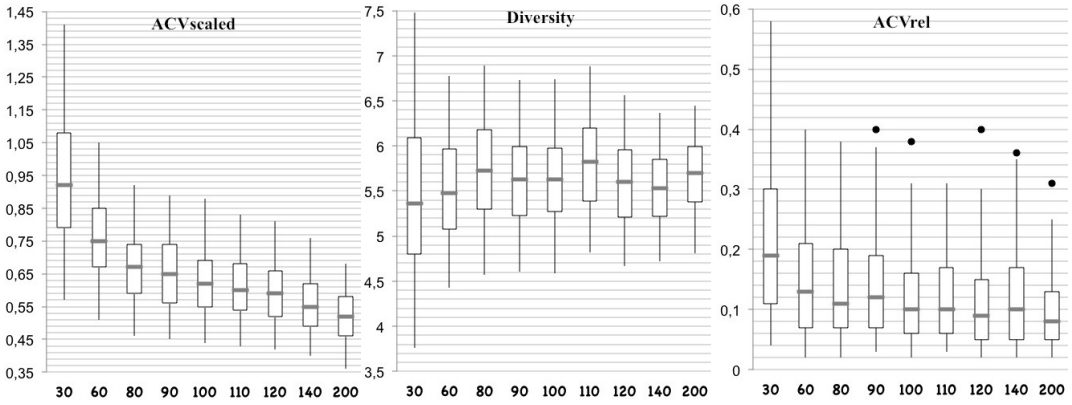

Fig. 6.137: 3D-Performance results of configurations with aggregate selection ($p_0 = 50\%$) and a variation of ps.

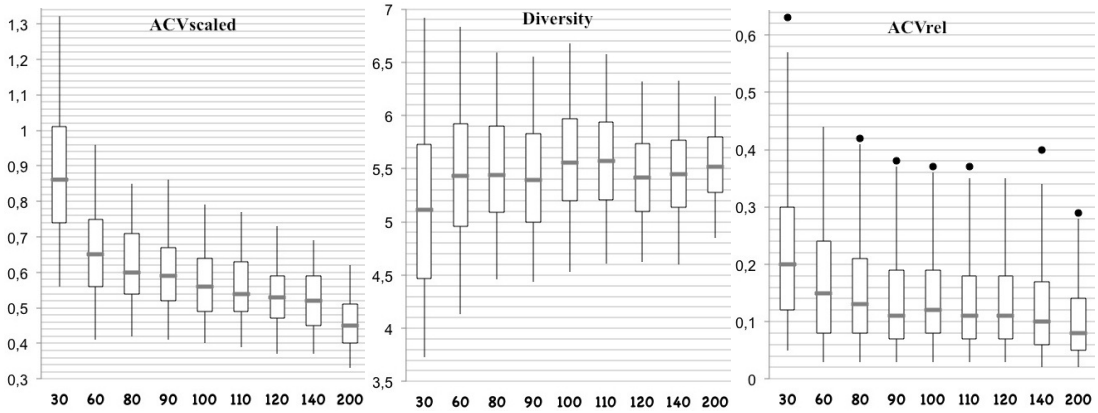

Fig. 6.138: 3D-Performance results of configurations with aggregate selection ($p_0 = 60\%$) and a variation of ps.

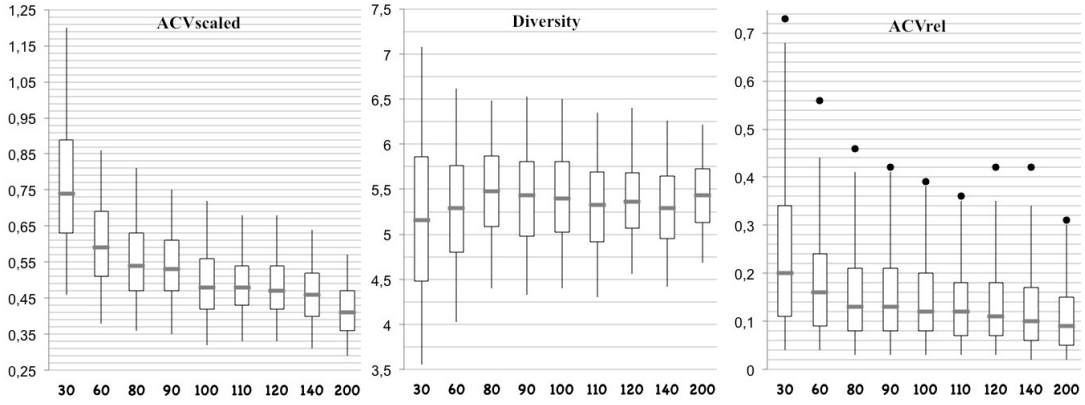

Fig. 6.139: 3D-Performance results of configurations with aggregate selection ($p_0 = 70\%$) and a variation of ps.

diversity results are reduced for ps values higher than 80 except for $ps = 110$, which is comparable to the diversity results of $ps = 80$. The highest diversity results are achieved for $ps = 80$ and 110, the lowest range of diversity values as well as a tendency towards lower values is achieved for $ps = 140$. The ACV_{rel}

values are continuously decreased by an increase of ps, only the performance of the configuration with $ps = 140$ is regarded as an outlier. Furthermore, the ACV_{rel} results reveal outlier for $ps = 90$ and higher ps values, revealing some remarkably higher indicator values.

The ACV_{scaled} and ACV_{rel} performance results are transferable to the configurations with $p_0 = 60\%$ (Figure 6.138, advisable p_0 setting according to the Section 6.2): The ACV_{scaled} and ACV_{rel} values reveal an improved performance by an increase of ps. The highest diversity performance is achieved for $ps = 100, 110$ and is nearly comparable to the configurations with $ps = 60$ and 80. The range of diversity results is decreased and the diversity values are in general lower for ps higher than 110 compared to the configurations with $ps = 60$ to 90.

The performance results of the configurations with $p_0 = 70\%$ reveal similar metric results of ACV_{scaled} and ACV_{rel}: The increase of ps results in a decrease of these indicator values and the range of diversity values (Figure 6.139). Diversity performance is more comparable for different ps values. The highest diversity values are achieved for $ps = 80$ and 100. Diversity is generally reduced for ps higher than 100.

In general, ACV_{scaled} performance tendency is reduced by an increase of p_0. Diversity performance is more comparable for higher p_0 settings and different values of ps. In contrast to the performance results of the configurations with ACV-based or ACV-random selection, the ACV_{rel} indicator values are not decreased by an increase of p_0 and therefore, the indicator values are in the same range for all p_0 settings.

The following series of tests are performed on the 4D-MOP and the confi-

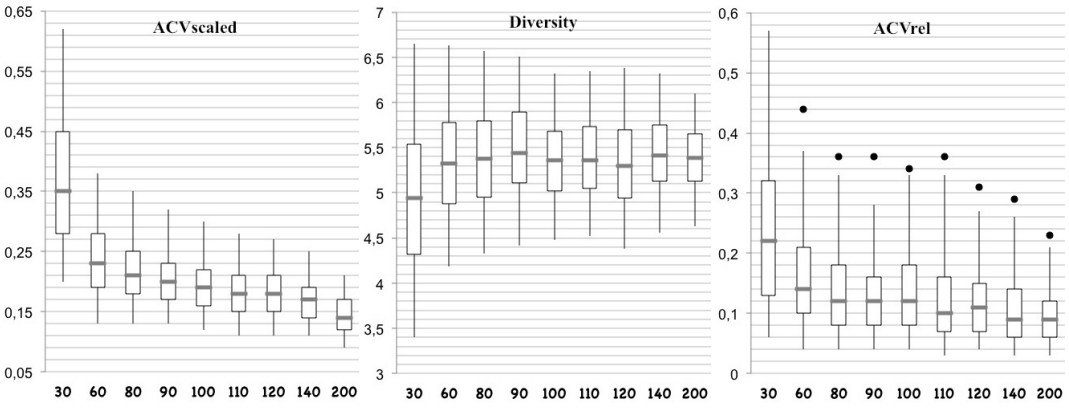

Fig. 6.140: 4D-Performance results of configurations with aggregate selection ($p_0 = 50\%$) and a variation of ps.

gurations with aggregate selection, different p_0 values and a variation of ps. In general, the performance results are once again comparable in their general

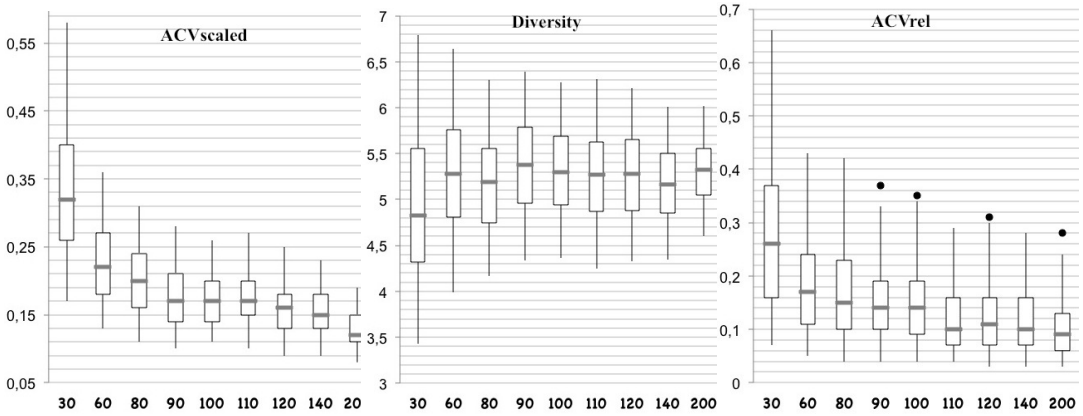

Fig. 6.141: 4D-Performance results of configurations with aggregate selection ($p_0 = 60\%$) and a variation of ps.

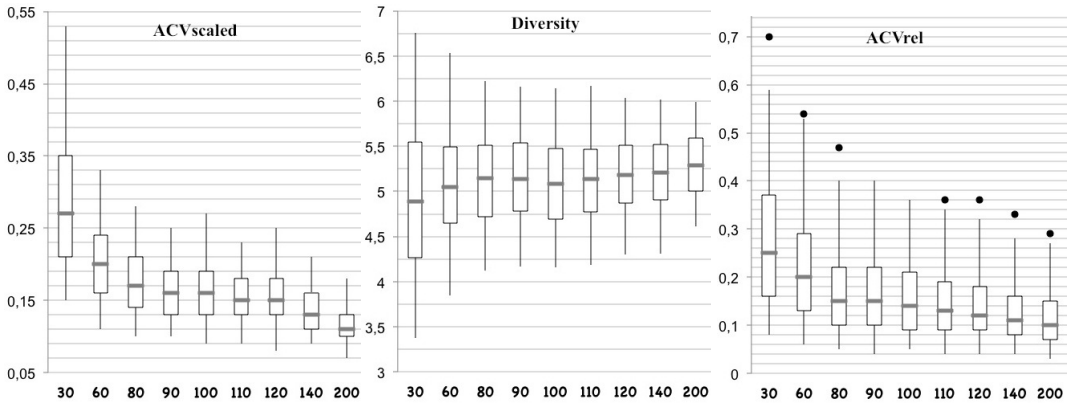

Fig. 6.142: 4D-Performance results of configurations with aggregate selection ($p_0 = 70\%$) and a variation of ps.

tendency to the performance results of the configurations by ps variation, as presented above: Figure 6.140 depicts the performance results of the configurations with $p_0 = 50\%$ and a variation of ps. The convergence performance is significantly improved by an increase of ps, the ACV_{scaled} results are nearly comparable for $ps = 100$ and $ps = 120$. The highest diversity results are achieved for $ps = 90$. The diversity results are more constant but tend to be slightly lower for ps values higher than $ps = 90$. The relative non-dominated solution quality is significantly improved by an increase of ps. The ACV_{rel} performance of the configurations with $ps = 80$ and 100 are comparable. Once again, the ACV_{rel} results are skewed and outliers are achieved nearly for every ps setting revealing a tendency towards higher ACV_{rel} results.

Figure 6.141 presents the performance results of the configurations with $p_0 = 60\%$, the advisable parameter setting and a variation of ps. The convergence results are generally improved by an increase of ps, the ACV_{scaled} results of $ps = 120$ and 140 are comparable as well as those of $ps = 100$ and 110. The

results of the configuration with $ps = 200$ are skewed with the tendency to lower ACV_{scaled} results. The tendency towards the highest diversity results is achieved for $ps = 90$ and the diversity performance of the configurations with $ps = 100$ to 120 is nearly comparable. The relative non-dominated solution quality is improved by an increase of ps. The ACV_{rel} results of $ps = 60$ and 80 are comparable to each other as well as the results of $ps = 90$ and 100 and those of $ps = 110$ to 140. The results are slightly skewed, but outliers are only achieved for $ps = 100, 120$ and 200.
Figure 6.142 presents the performance results of the configurations with $p_0 = 70\%$ and a variation of ps. The convergence performance is continuously improved by an increase of ps. The fluctuations of the diversity values are significantly decreased by an increase of ps. The ACV_{scaled} results of $ps = 90$ and 100 as well as those of $ps = 110$ and 120 are comparable. Furthermore, the tendency to higher diversity results is more equivalent for all ps settings compared to the diversity performance of configurations with $p_0 = 50\%$ and $p_0 = 60\%$. The relative non-dominated solution quality is improved by an an increase of ps, but the results are once again slightly skewed with a tendency towards remarkably higher ACV_{rel} values also visible by the outliers.
In general, the convergence behavior as well as the relative non-dominated solution quality is comparable to the performance results of the corresponding 3D-results (Figure 6.137 - Figure 6.139) with regard to the general performance tendency of the results by an increase of ps: The ACV_{scaled} and ACV_{rel} results are decreased in the span as well as with regard to the absolute indicator values. The ACV_{rel} indicator values are almost in the same range independent of the p_0 settings and different ps settings. This is different to the performance results of configurations with ACV-based and ACV-random, where the span is reduced by an increase of p_0 with regard to the absolute indicator values. Since the convergence performance and the relative non-dominated solution quality is continuously decreased by an increase of ps, but diversity reveals no significant improvement by an increase of ps above 110, the advisable range of ps settings is in the range of 90 to 110. Once again, the advisable range of ps settings is a compromise between improved performance results and computational complexity, which is significantly increased for higher ps settings.

6.4.4 Discussion

The test series resented in this section provide an insight into the interdependence between the selection strategies and the population size. The test results

reveal some general observations for all selection strategies independent of the p_0 settings: The range of the three indicator values is remarkably reduced by an increase of *ps* exposing smaller deviations of the indicator results. The convergence and relative non-dominated solution quality is improved by an increase of *ps* for all selection strategies and independent of the selection parameter p_0. The level of improvement is reduced for *ps* settings higher than 100. The tendency towards higher diversity values is also reduced for *ps* settings higher than 100.

The results matrix 6.11 presents the 2σ-confidence limits of the performance

ps	ind.	ACV-based, 50%	ACV-random, 10%	AGG selection, 60%
30	ACV_{scaled}	[0.78, 0.96]	[0.85, 1,03]	[0.80, 0.98]
	Diversity	[4.88, 5.84]	[4.96, 5.83]	[4.74, 5.60]
	ACV_{rel}	[0.07,0.18]	[0.15,0.28]	[0.16,0.30]
60	ACV_{scaled}	[0.61, 0.72]	[0.66, 0.78]	[0.60, 0.72]
	Diversity	[5.01, 5.68]	[5.15, 5.81]	[5.13, 5.78]
	ACV_{rel}	[0.06,0.13]	[0.10,0.20]	[0.12,0.23]
80	ACV_{scaled}	[0.57, 0.67]	[0.59, 0.70]	[0.57, 0.67]
	Diversity	[5.19, 5.76]	[4.15, 5.72]	[5.23, 5.77]
	ACV_{rel}	[0.04,0.11]	[0.09,0.18]	[0.11,0.21]
90	ACV_{scaled}	[0.55, 0.65]	[0.57, 0.66]	[0.55, 0.65]
	Diversity	[5.27, 5.88]	[5.18, 5.68]	[5.17, 5.69]
	ACV_{rel}	[0.05,0.12]	[0.09,0.18]	[0.09,0.18]
100	ACV_{scaled}	[0.53, 0.61]	[0.55, 0.65]	[0.53, 0.62]
	Diversity	[5.30, 5.82]	[5.31, 5.82]	[5.33, 5.83]
	ACV_{rel}	[0.04,0.10]	[0.08,0.16]	[0.10,0.18]
110	ACV_{scaled}	[0.50, 0.58]	[0.54, 0.63]	[0.52, 0.60]
	Diversity	[5.36, 5.85]	[5.28, 5.77]	[5.34, 5.84]
	ACV_{rel}	[0.04,0.09]	[0.08,0.16]	[0.10,0.18]
120	ACV_{scaled}	[0.51, 0.60]	[0.55, 0.64]	[0.49, 0.57]
	Diversity	[5.25, 5.73]	[5.42, 5.90]	[5.19, 5.67]
	ACV_{rel}	[0.04,0.10]	[0.08,0.16]	[0.09,0.17]
140	ACV_{scaled}	[0.47, 0.55]	[0.51, 0.59]	[0.48, 0.56]
	Diversity	[5.31, 5.75]	[5.25, 5.69]	[5.25, 5.68]
	ACV_{rel}	[0.03,0.08]	[0.06,0.13]	[0.07,0.18]
200	ACV_{scaled}	[0.43, 0.50]	[0.46, 0.52]	[0.43, 0.49]
	Diversity	[5.37, 5.72]	[5.31, 5.63]	[5.37, 5.71]
	ACV_{rel}	[0.03,0.07]	[0.06,0.12]	[0.07,0.14]

Table 6.11: Results matrix presenting convergence, diversity and relative non-dominated solution quality achieved by different selection configurations with advisable p_0 settings and a variation of *ps* in the case of the 3D-MOP.
Backgrounds: improving indicator results (green), optimal indicator results (red), constant indicator results (yellow), medium performance (blue).

means obtained by the configurations with the three selection strategies and the advisable selection parameter settings p_0 according to Section 6.2, in the case of the 3D-MOP. The 2σ-confidence limits of the 4D-performance means are depicted in Table 6.12. The green highlighted intervals identify continuously improving indicator results, the red highlighted intervals indicate the best

ps	ind.	ACV-based, 50%	ACV-random, 10%	AGG selection, 60%
30	ACV_{scaled}	[0.27, 0.37]	[0.31, 0,40]	[0.20, 0.38]
	Diversity	[4.35, 5.14]	[4.87, 5.67]	[4.55, 5.37]
	ACV_{rel}	[0.08,0.21]	[0.16,0.29]	[0.21,0.35]
60	ACV_{scaled}	[0.20, 0.25]	[0.21, 0.27]	[0.20, 0.25]
	Diversity	[4.81, 5.48]	[4.98, 5.67]	[4.97, 5.63]
	ACV_{rel}	[0.05,0.11]	[0.11,0.20]	[0.14,0.23]
80	ACV_{scaled}	[0.17, 0.21]	[0.19, 0.24]	[0.18, 0.22]
	Diversity	[4.87, 5.44]	[4.97, 5.58]	[4.92, 5.48]
	ACV_{rel}	[0.04,0.09]	[0.09,0.16]	[0.13,0.22]
90	ACV_{scaled}	[0.16, 0.20]	[0.18, 0.22]	[0.16, 0.20]
	Diversity	[4.87, 5.48]	[5.03, 5.57]	[5.12, 5.64]
	ACV_{rel}	[0.05,0.12]	[0.09,0.16]	[0.12,0.19]
100	ACV_{scaled}	[0.15, 0.19]	[0.17, 0.21]	[0.15, 0.19]
	Diversity	[4.88, 5.35]	[5.04, 5.51]	[5.08, 5.57]
	ACV_{rel}	[0.03,0.08]	[0.08,0.14]	[0.11,0.19]
110	ACV_{scaled}	[0.15, 0.18]	[0.16, 0.19]	[0.16, 0.19]
	Diversity	[4.94, 5.38]	[5.07, 5.57]	[5.01, 5.52]
	ACV_{rel}	[0.03,0.07]	[0.08,0.15]	[0.09,0.15]
120	ACV_{scaled}	[0.14, 0.18]	[0.16, 0.19]	[0.14, 0.18]
	Diversity	[4.88, 5.30]	[4.97, 5.42]	[5.04, 5.52]
	ACV_{rel}	[0.03,0.08]	[0.07,0.11]	[0.09,0.15]
140	ACV_{scaled}	[0.13, 0.16]	[0.15, 0.18]	[0.14, 0.17]
	Diversity	[4.90, 5.31]	[4.95, 5.36]	[4.97, 5.39]
	ACV_{rel}	[0.03,0.07]	[0.07,0.12]	[0.09,0.15]
200	ACV_{scaled}	[0.12, 0.14]	[0.13, 0.15]	[0.12, 0.14]
	Diversity	[4.93, 5.31]	[5.09, 5.46]	[5.14, 5.48]
	ACV_{rel}	[0.02,0.05]	[0.05,0.10]	[0.08,0.13]

Table 6.12: Results matrix presenting convergence, diversity and relative non-dominated solution quality achieved by different selection configurations with advisable p_0 settings and a variation of ps in the case of the 4D-MOP.
Backgrounds: improving indicator results (green), optimal indicator results (red), constant indicator results (yellow), medium performance (blue).

achieved indicator results, the yellow ones mark the constant indicator results and the blue one the intervals positioned in a medium performance range. The result matrix 6.11 reveals a continuous improvement of the convergence by an increase of ps, indicated by the green highlighted ACV_{scaled} confidence limits in the case of all three selection strategies. A great proportion of optimal diversity and ACV_{rel} results are achieved for ps settings higher than 100, indicated by the high number of red highlighted intervals in Table 6.11. The relative non-dominated solution quality achieves a considerable number of constant performance results in the range of $ps = 100$ to 120 for all three selection strategies, indicated by the high amount of yellow highlighted intervals. Since the improvement of convergence and relative non-dominated solution quality is significantly at the cost of diversity for ps settings higher than 110, the range of $ps = 90$ to $ps = 110$ is regarded as the advisable range of ps settings for all three selection strategies.

The results matrix 6.12 presents the 2σ-confidence limits of the performance means in the case of the 4D-MOP. Such as in the case of the 3D-MOP, convergence is continuously improved by an increase of ps indicated by the green highlighted ACV_{scaled} intervals. Optimal diversity is generally achieved for the three selection strategies in a range of $ps = 80$ to 110. Moreover, the relative non-dominated solution quality is also improved by an increase of ps, but provides some constant results from $ps = 100$ on. Since the improvement of convergence and relative non-dominated solution quality is significantly at the cost of diversity for ps settings higher than 110 and the ACV_{rel} results are nearly constant for $ps = 100$ to $ps = 140$, the advisable range of ps settings is regarded as $ps = 90$ to $ps = 110$ for all three selection strategies.
In the following, the questions raised in the first part of this section are answered: Firstly, the investigation goal is aimed at the influence of large populations on the convergence speed. Early convergence as a main goal of VONSEA is defeated since an increase of the population size results in higher speed of convergence. There is a continuous convergence improvement observable by an increase of ps, but the level of improvement is reduced for higher ps values at the cost of diversity, as demonstrated by the experimental results in this section. Secondly, the investigations are analyzed regarding an impact of the population size and the selection parameter. A configuration rule for the selection parameter depending on the population size is necessary in the case of a large interdependence of both. However, the experiments do not reveal an interdependence of the population size and the selection parameter in general. An advisable range of ps values providing optimal performance is set in the range of 90 to 110 for the 3D-MOP as well as 4D-MOP and independent of the selection strategy.

6.5 Comparison of VONSEA and NSGA-II

The last experimental section presents the performance differences between VONSEA and the character-encoded state-of-the-art NSGA-II with a diversity preserving selection strategy in the case of the 3D- and 4D-MOP. The comparison is performed under practical points of view and gives an insight into the different search properties of VONSEA and NSGA-II. These algorithms are compared to each other regarding convergence and diversity properties and are exemplarily analyzed on the basis of the achieved non-dominated solutions detected in the first five generations. From a practical point of view, an algorithm has to provide a large number of diverse and maximally qualified

solutions in each generation since these candidate peptides will be synthesized and analyzed in the subsequent *in vitro* drug design step. For this purpose, the non-dominated solutions of the first five generations provided by NSGA-II and VONSEA are analyzed regarding the quantity of high quality solutions, the quality range within the non-dominated solutions and the detection progress of high quality solutions over the first generations. Moreover, the number of identified non-dominated solutions by VONSEA and NSGA-II are evaluated according to the average number of dominated solutions per solution in each other's set of non-dominated solutions for a deeper insight into the abilities of VONSEA.

6.5.1 Simulation Settings

	NSGA-II	**VONSEA**
population size generations	100 5	100 5
recombination	one-point crossover parent number: 2 $p_c = 1.0$	LiDeRP parent number: 3
mutation	one-point mutation $p_m = 1.0$	BS_{adap}
selection	diversity preserving method 1. the 10 best solutions are selected in P_{t+1} 2. binary tournament selection: selection criterion: crowding distance	ACV-based $p_0 = 50\%$ $ts = 10$

Table 6.13: Algorithm settings of NSGA-II and VONSEA for the performance comparison

Table 6.13 gives an overview of the algorithm settings. The procedure of NSGA-II used for the experiments is kept similar to the traditional NSGA-II, whereas the individuals for the succeeding generations are selected by a combination of elitism and diversity preserving method: One-point recombination is used, whereby the recombination points are selected randomly. Furthermore, two parent are selected for recombination according to the traditional procedure. The recombination probability is set to 1.0. Afterward, each solution is mutated by one-point mutation. Once again, the mutation points are chosen randomly. The selection strategy of the traditional NSGA-II has been adapted to preserve diversity in the population. Firstly, the 10 best solutions of the combined

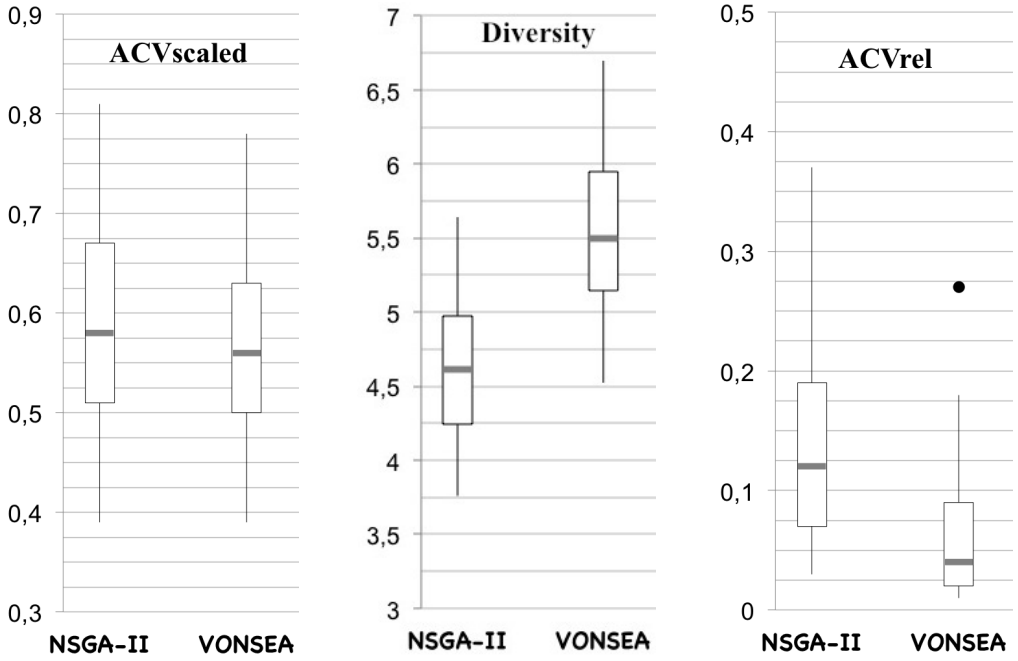

Fig. 6.143: 3D-Performance results of NSGA-II and VONSEA.

parent and offspring population are selected into the succeeding generation. The remaining individuals for the succeeding generation are selected by binary tournament selection and the individual with the higher crowding distance value is chosen into the succeeding generation. The settings of VONSEA are the advisable components and parameter values, as stated in the preceding sections.

The total number of non-dominated solutions identified by VONSEA and NSGA-II in each generation are evaluated by comparing the average number of dominated solutions per solution in each other's set of non-dominated solutions for a deeper insight into the abilities of VONSEA. The following indicator is used for this comparison:

$$avg(F_1, F_2) = \frac{(\sum_{i=1}^{|F_1|} n_i)/|F_2|}{m} \quad \text{with} \tag{6.1}$$

$$n_i = \{b \in F_2 |\ a_i \prec b \text{ with } a_i \in F_1\} \quad \text{and} \quad m = \{a \in F_1 |\ \exists b \in F_2 : a \prec b\},$$

where F_1 and F_2 are two sets of non-dominated solutions that are to be compared. The sum of dominated solutions in F_2 per solution in F_1 is set in relation to the total number of solutions in F_2 to ensure the independence of this indicator from the cardinality. This indicator is not symmetric and both values of $avg(F_1, F_2)$ and $avg(F_2, F_1)$ have to be determined. A value of $avg(F_1, F_2) = 1$ implicates that F_1 strongly dominates F_2 and a values of $avg(F_1, F_2) = 0$ indicates that F_2 strongly dominates F_1.

6.5.2 Results of the 3D-MOP

Figure 6.143 presents the convergence, diversity and relative non-dominated solution quality of NSGA-II and VONSEA in the case of the 3D-MOP. NSGA-II generally reveals similar ACV_{scaled} values with a tendency towards higher convergence indicator values compared to VONSEA. Diversity of NSGA-II is drastically reduced compared to VONSEA, though the individuals for the succeeding generations are selected by a diversity preserving method. The relative non-dominated solution quality is remarkably high compared to the corresponding ACV_{rel} results of VONSEA, revealing a general tendency towards non-dominated solutions with a remarkably lower quality compared to those of VONSEA. This conclusion is obvious since the ACV_{rel} values are calculated using the ACV values of the non-dominated solutions relative to the ACV values of the entire population, which are nearly comparable in the case of VONSEA and NSGA-II. The following analysis of the non-dominated solutions within the first five generations, detected by VONSEA and NSGA-II, underlines this conclusion.

Figures 6.144 - 6.148 represent the bar graphs with error indicators (standard deviation) of the averaged three molecular function values obtained by the non-dominated solutions of NSGA-II and VONSEA in the first five generations. Standard deviations are depicted by the whiskers. The objective function values of Hydro have been scaled for a better visualization. Figure 6.144 reveals that the three objective functions values of the non-dominated solutions detected by VONSEA are generally better and in the case of MW and Hydro even significantly better under the condition that the objective functions have to be minimized. Standard deviations are higher for the three objective functions values achieved by NSGA-II compared to VONSEA revealing a wide quality spread of the objective function values and more abstract, of non-dominated solutions. Figure 6.145 presents the mean values plot of the objective functions values from the non-dominated solutions detected by VONSEA and NSGA-II in the second generations. The objective values of MW and Hydro are on average considerably lower in the case of VONSEA compared to NSGA-II, similar to the standard deviations. Only the NMW objective values of VONSEA are slightly higher than those of NSGA-II. The objective NMW is used as a similarity measure and the NMW values therefore allow an insight into the genetic diversity of the solutions. The higher NMW values achieved by VONSEA are potentially a consequence of the higher genetic diversity within the non-dominated solutions caused by the considerably higher mutation and recombination probability of VONSEA in the first five generations. Generally,

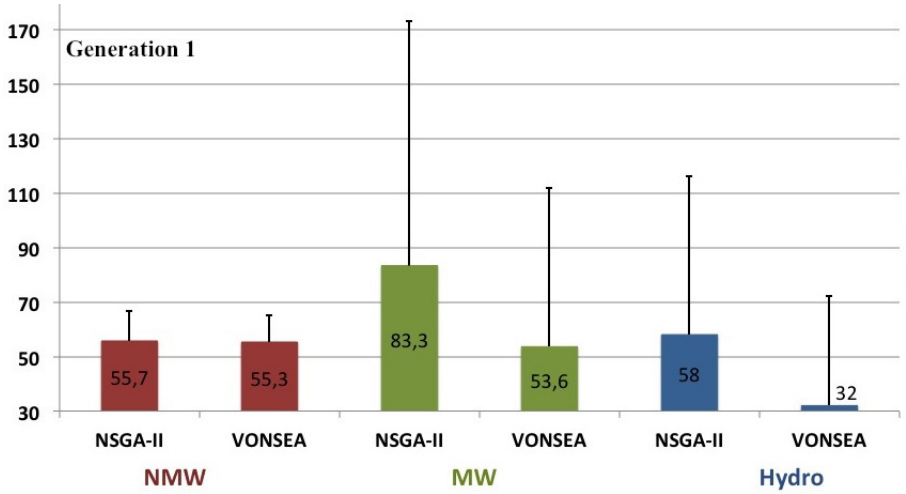

Fig. 6.144: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the first generation. (3D-MOP)

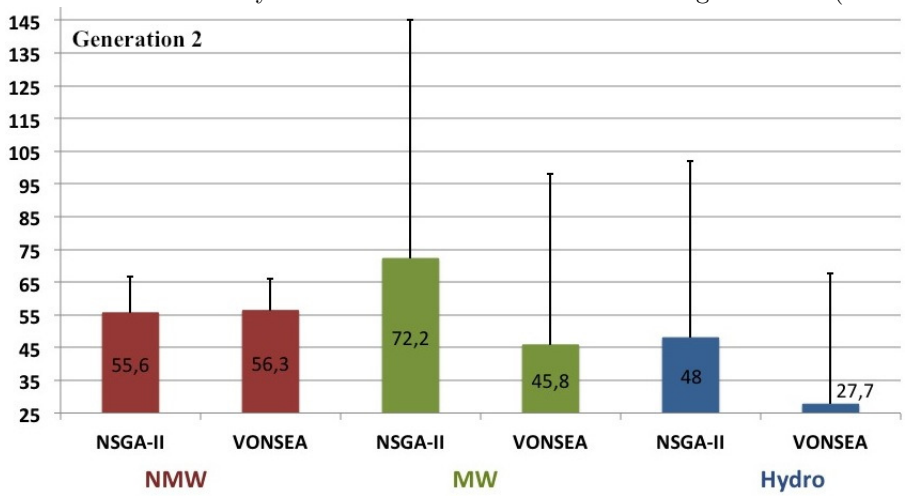

Fig. 6.145: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the second generation. (3D-MOP)

all objective functions values - except NMW in the case of VONSEA - are on average improved from the first to the second generation, indicating that both algorithms proceeds in the term convergence. The averaged objective values of non-dominated solutions in the third generation detected by VONSEA are furthermore remarkably better than those of NSGA-II except in the case of NMW values. These are stagnating compared to the last generation in the case of VONSEA (Figure 6.147). The standard deviations of NSGA-II are also higher than those of VONSEA revealing once again a wide quality spread of the non-dominated solutions. NSGA-II remarkably proceeds in the term of convergence, but is not able to outperform the performance results of VONSEA. In the case of VONSEA, the NMW values of the non-dominated solutions are stagnated, those of MW are on average slightly higher and those of Hydro are

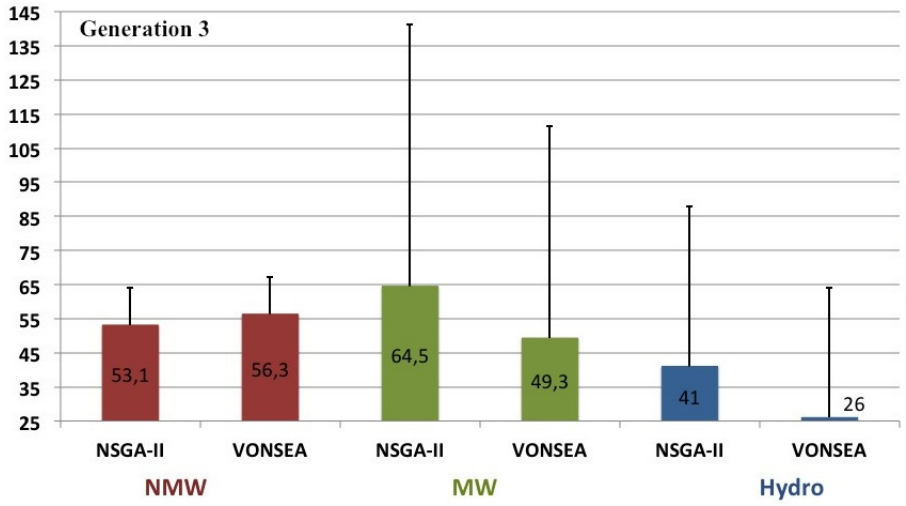

Fig. 6.146: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the third generation. (3D-MOP)

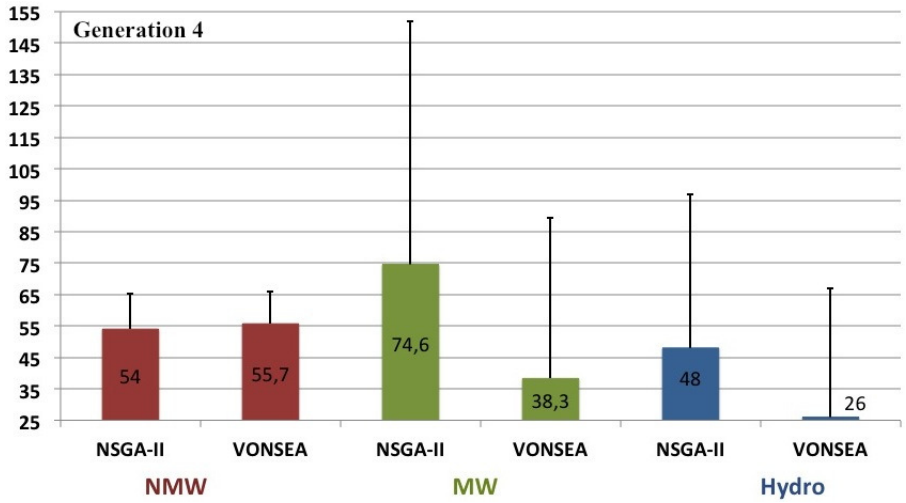

Fig. 6.147: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the fourth generation. (3D-MOP)

improved compared to the results of the second generation. The results of the fourth generation reveal a similar pattern (Figure 6.147): The average objective values of VONSEA are considerably better for MW and Hydro compared to those of NSGA-II, the NMW values of VONSEA are slightly higher than those of NSGA-II revealing a higher genetic diversity within the non-dominated solutions of VONSEA. This time, NSGA-II achieves worse averaged objective values for MW and Hydro, whereas VONSEA reveals improved averaged objective values for NMW and MW and stagnated values for Hydro. Furthermore, the averaged objective values of the non-dominated solutions detected by VONSEA are remarkably better in the case of MW and Hydro compared to NSGA-II in the fifth generation (Figure 6.148). The NMW objective

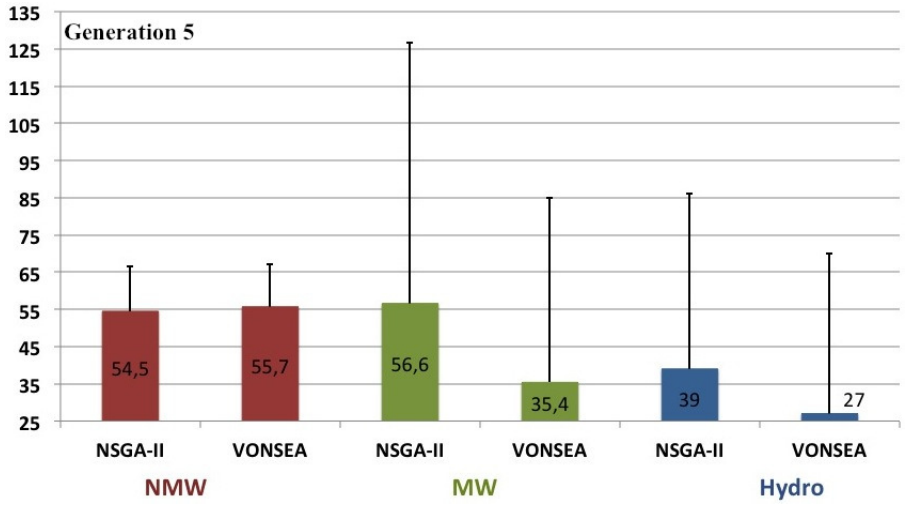

Fig. 6.148: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the fifth generation. (3D-MOP)

values of VONSEA are once again stagnated compared to those of the last generation. Also the standard deviations of NSGA-II are higher than those of VONSEA, revealing less stable results. The convergence performance regarding the non-dominated solutions of NSGA-II achieves the highest progress except in the case of the NMW values, which are slightly higher. VONSEA achieves a progress in the objective MW, stagnating performance for NMW and slightly increased values on average for Hydro.

Figure 6.149 presents the spread of the number of non-dominated solutions

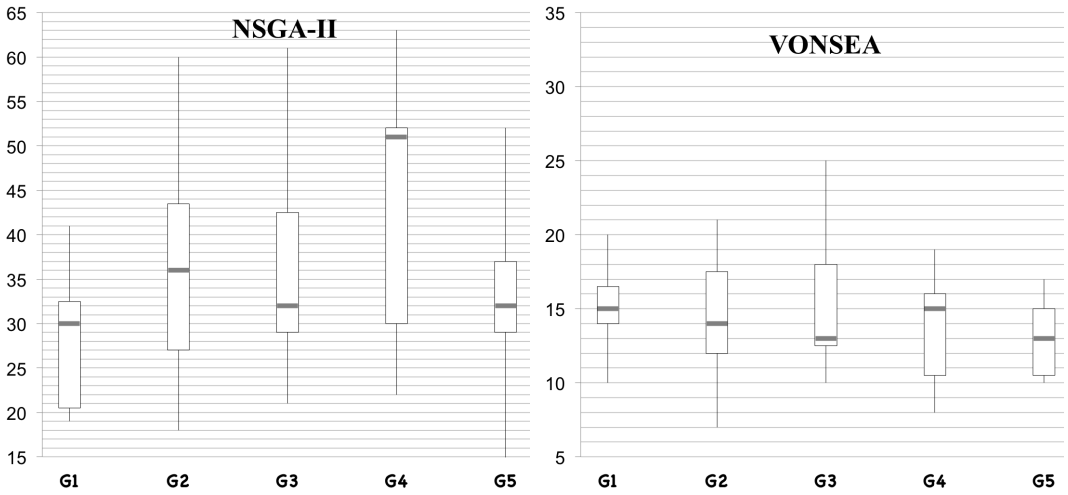

Fig. 6.149: Number of non-dominated solutions provided by NSGA-II (left figure) and VONSEA (right figure) in the first five generations (G1)-(G5). (3D-MOP)

detected by NSGA-II and VONSEA in the first five generations. NSGA-II generally achieves a considerably higher number of non-dominated solutions.

The first generation achieves the lowest number of non-dominated solutions followed by the fifth generation. The higher number of non-dominated solutions is achieved in the fourth generation, whereas the boxplot reveals skewed results with a clear tendency towards a higher solution number. The number of non-dominated solutions achieved by VONSEA is generally in the same numerical range, which is significantly lower than in the case of NSGA-II. The boxplots of generation three and four reveal that the number of solutions is strongly skewed. Furthermore, the number of solutions is continuously reduced in the generations four and five. These results and the results of the mean value plots allow the following conclusions: VONSEA provides a significantly lower but nearly reliable number of non-dominated solutions, which are on average of a considerably higher quality and a slightly higher genetic diversity than NSGA-II. On the contrary, NSGA-II achieves a high number of non-dominated solutions, which have a wide quality range. From the practical point of view, VONSEA provides an adequate number of high quality peptides in each of the first five generation, whereas NSGA-II provides a wide range of currently optimal peptides, which require a further thorough selection.
Figure 6.150 gives a more precise insight into the quality of the non-dominated solutions detected in the test runs. The quality is evaluated according to the average number of dominated solutions per solution in each other's set of non-dominated solutions per generation using the indicator (6.1). The figure presents a performance superiority of the non-dominated solutions of VONSEA in each generation compared to those of NSGA-II, since the indicator values avg(VONSEA, NSGA-II) are significantly higher than avg(NSGA-II, VONSEA) in each generation. This indicates that each non-dominated solution detected by VONSEA dominates on average a considerably high number of non-dominated solutions detected by NSGA-II. As a consequence, VONSEA provides a lower and chosen number of highly qualified candidate peptides, whereas NSGA-II detects a large number of candidate peptides revealing a wide quality range. Therefore, the set of non-dominated solutions detected by NSGA-II needs a further target-oriented sorting. This allows the conclusion that the search performance of VONSEA is more efficient and target-orientated than the one of NSGA-II.

6.5.3 Results of the 4D-MOP

In the following, NSGA-II and VONSEA performance is compared in the case of the 4D-MOP. Figure 6.157 depicts convergence, diversity and relative

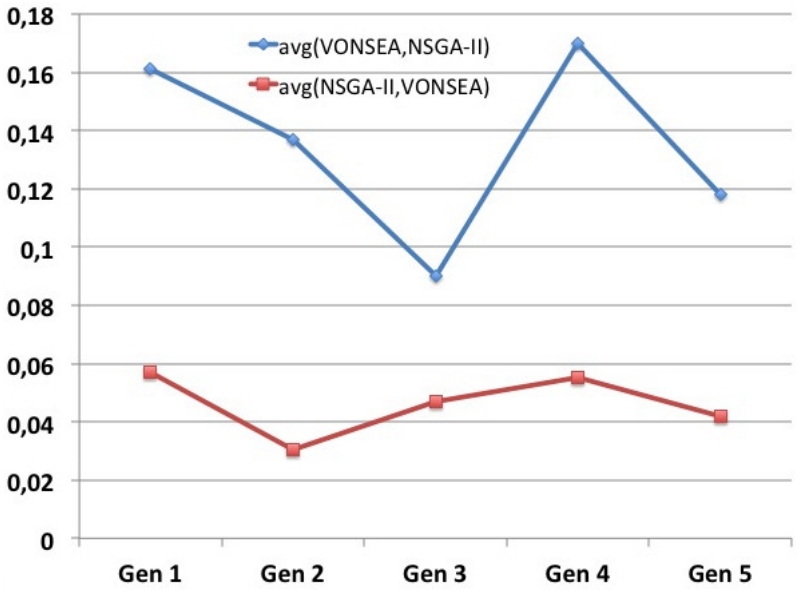

Fig. 6.150: Comparison of the detected non-dominated solutions of VONSEA and NSGA-II per generation by the average number of dominated solutions per solution in each other's set of non-dominated solutions. (3D-MOP)

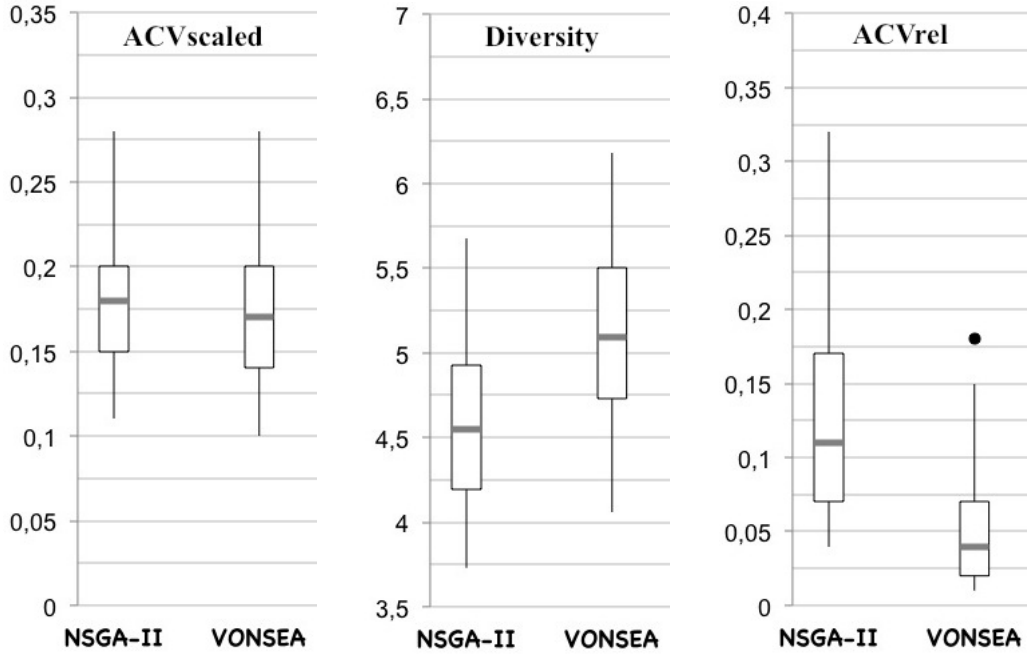

Fig. 6.151: 4D-Performance results of NSGA-II and VONSEA.

non-dominated solution quality of the NSGA-II and VOSEA configuration as described in Table 6.13. The ACV_{scaled} values of NSGA-II are compared to those of VONSEA with a slight tendency towards higher values. Otherwise, the diversity of NSGA-II is significantly lower compared to VONSEA. Furthermore, the ACV_{rel} results of NSGA-II are considerably higher than the results

of VONSEA. Once again, the ACV_{scaled} values of NSGA-II and VONSEA are generally comparable. This allows the conclusion that the non-dominated solutions of NSGA-II are of a lower quality than those of VONSEA. The following non-dominated solution analysis of the first five generations in the case of VONSEA and NSGA-II underlines this conclusion.

Figure 6.152 - 6.156 represent the bar graphs with error indicators (stan-

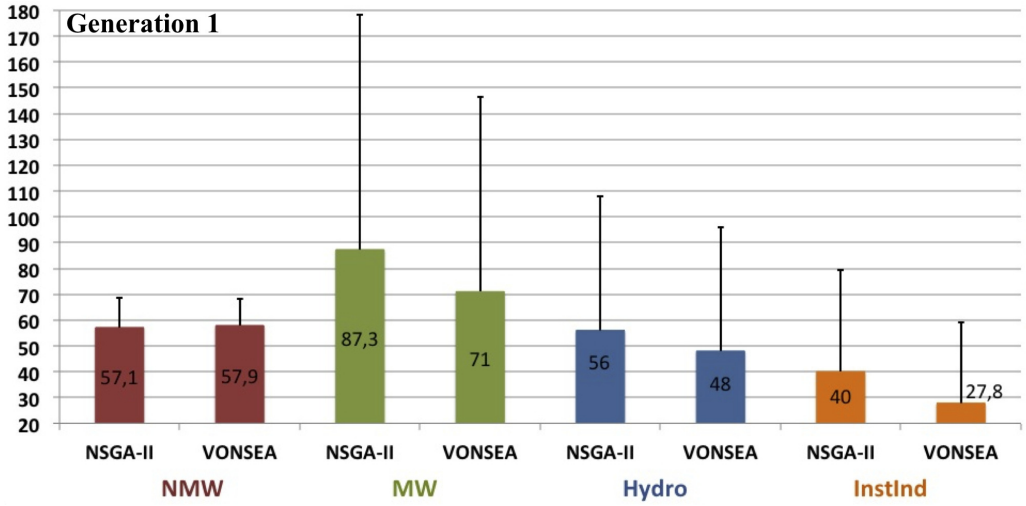

Fig. 6.152: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the first generation. (4D-MOP)

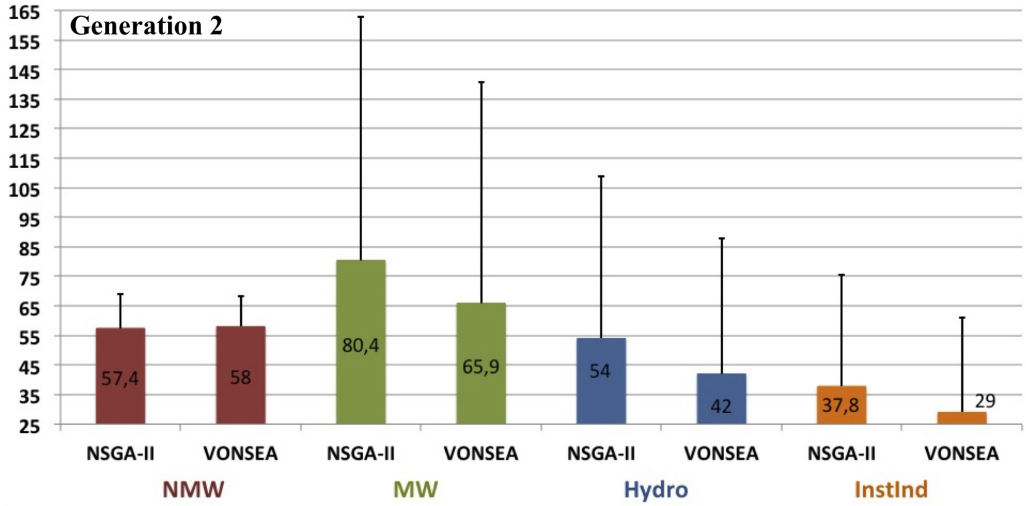

Fig. 6.153: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the second generation. (4D-MOP)

dard deviation) of the averaged tour molecular function values obtained by the non-dominated solutions of NSGA-II and VONSEA in the first five generations. The objective function values of Hydro and InstInd have been scaled for a better visualization. The comparison of the non-dominated objective function values of NSGA-II and VONSEA in the first generation reveals that

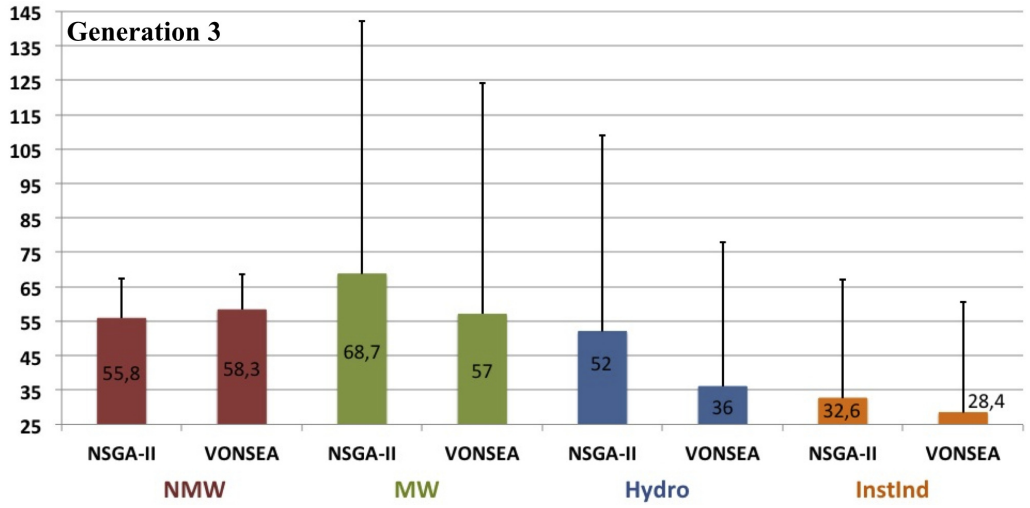

Fig. 6.154: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the third generation. (4D-MOP)

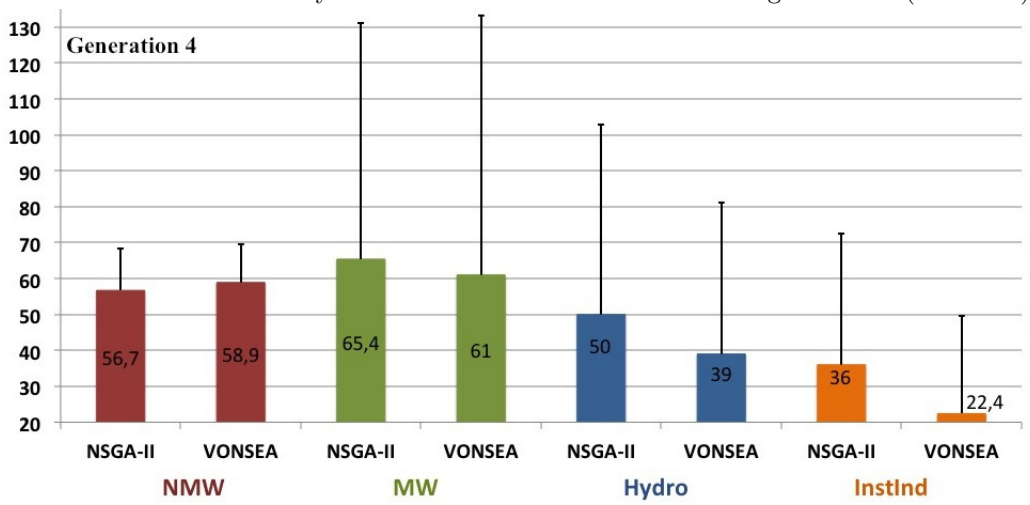

Fig. 6.155: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the fourth generation. (4D-MOP)

VONSEA provides non-dominated solutions with remarkably better objective values for MW, Hydro, InstInd and comparable NMW function values compared to NSGA-II (Figure 6.152). Furthermore, the standard deviation of the objective function values achieved by NSGA-II is generally higher indicating a wide range of objective values. Therefore, the peptides provided by VONSEA outperform those of NSGA-II on average in three objective function values. Furthermore, the range of the peptide quality is generally lower in the case of VONSEA. The non-dominated solutions provided by VONSEA in the second generation also outperform those of NSGA-II on average in the objective values of MW, Hydro and InstInd (Figure 6.153). Otherwise, the objective values of NMW are on average slightly higher in the case of VONSEA compared to

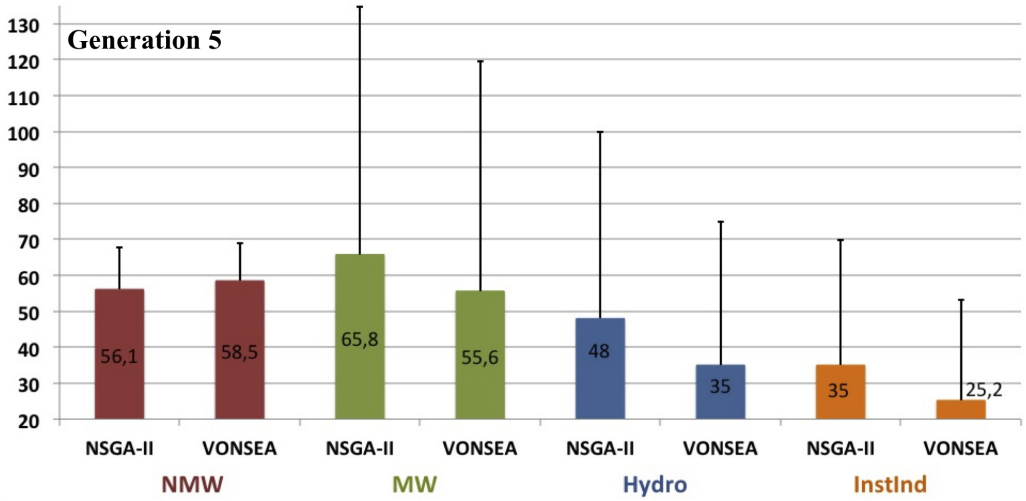

Fig. 6.156: Average objective function values and standard deviation of the non-dominated solutions achieved by NSGA-II and VONSEA in the fifth generation. (4D-MOP)

NSGA-II revealing a tendency towards a higher genetic diversity. The standard deviation of NSGA-II is also higher for all objective values compared to VONSEA. Compared to the results of the first generation, the non-dominated solutions of NSGA-II and VONSEA progresses in the three objective values MW, Hydro and InstInd, whereas the genetic diversity slightly increases from the first to the second generation in the case of VONSEA and NSGA-II. A similar pattern is achieved by the results of the third generation (Figure 6.154): The non-dominated solutions of VONSEA outperform those of NSGA-II in the objectives MW, Hydro and InstInd, whereas the objective values of NMW reveal a higher genetic diversity of the non-dominated solution in the case of VONSEA. Moreover, the standard deviations achieved by NSGA-II are once again higher than those of VONSEA. A progress regarding the improvement of the non-dominated solutions in the third generation compared to the preceding generation is observable for all objective functions values in the case of NSGA-II. Also the non-dominated solutions of VONSEA are improved in three of the four objective function values, but the NMW values are on average increased compared to the results of the second generations, revealing a higher genetic diversity. VONSEA also outperforms NSGA-II in the three objective MW, Hydro and InstInd in the fourth generation (Figure 6.155). Once again, NMW values of VONSEA are on average higher than those of NSGA-II. Standard deviations of NSGA-II are higher for at least three objectives; only the standard deviation of the MW objective is higher in the case VONSEA. The progress of the non-dominated solutions proceeds in two of four objectives in the case on NSGA-II compared to the preceding generation, whereas

VONSEA reveals a reduction of the objective function values in three cases. However, the provided peptides of VONSEA are on average of a higher quality and of a higher genetic diversity compared to those of NSGA-II. This pattern is also observable by the results of the fifth generation (Figure 6.156): VONSEA provides non-dominated solutions with better objective function values on average, only the NMW values are once again higher compared to NSGA-II. The standard deviations are all higher in the case of NSGA-II, whereas the difference between the standard deviations of NSGA-II and VONSEA are reduced. A progress regarding the non-dominated solution quality on average is achieved in the three objective functions NMW, Hydro and InstInd in the case of NSGA-II compared to the preceding generation, whereas the values of MW are slightly decreased. Also VONSEA improved the non-dominated solution quality in the objective functions NMW, MW and Hydro, whereas those of InstInd are reduced.

Figure 6.157 compares the number of non-dominated solutions achieved by

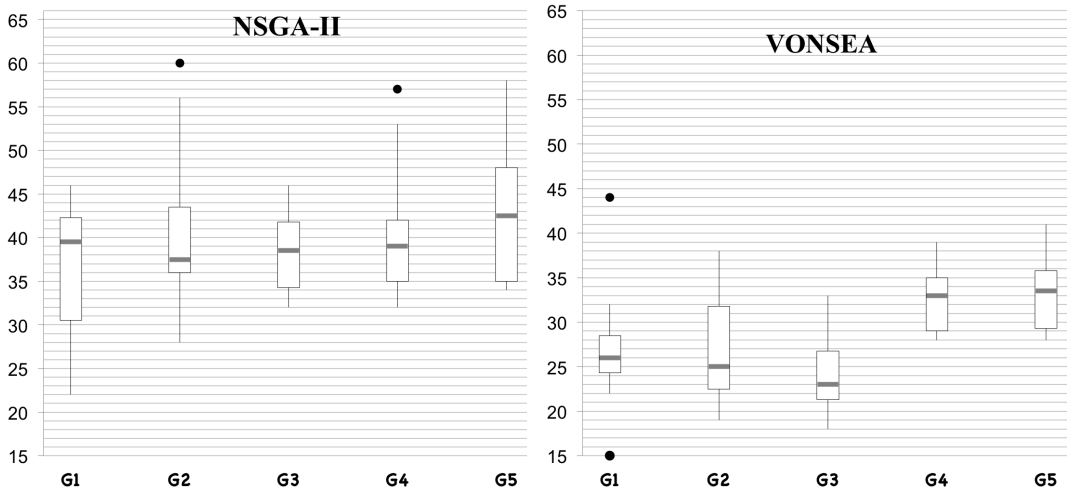

Fig. 6.157: Number of non-dominated solutions provided by NSGA-II (left figure) and VONSEA (right figure) in the first five generations (G1)-(G5). (4D-MOP)

NSGA-II and VONSEA in the first five generations. The lowest number of non-dominated solutions in general is provided by NSGA-II in the first generation, the highest number on general in the fifth generation. The results of the first two generations are skewed with a tendency towards higher values in the case of the first generation and a tendency towards lower values in the case of the second generation, observable by the median values. The number of provided non-dominated solutions is remarkably lower in the case of VONSEA. In the first three generations, the number of solutions is continuously decreased on average, whereas the results in the second generation are skewed, revealing a tendency towards a lower number of non-dominated solutions. Af-

ter the third generation, a significant leap is observable by a strong rise of the solution number. As a consequence, VONSEA provides generally a low but well-selected number of high quality non-dominated solutions, which are on average of a better quality than those provided by NSGA-II. Furthermore, VONSEA outperforms NSGA-II regarding the proposed 'best' peptide candidates in the case of the 3D-MOP and 4D-MOP.

Figure 6.158 gives a more precise insight into the quality of the non-dominated

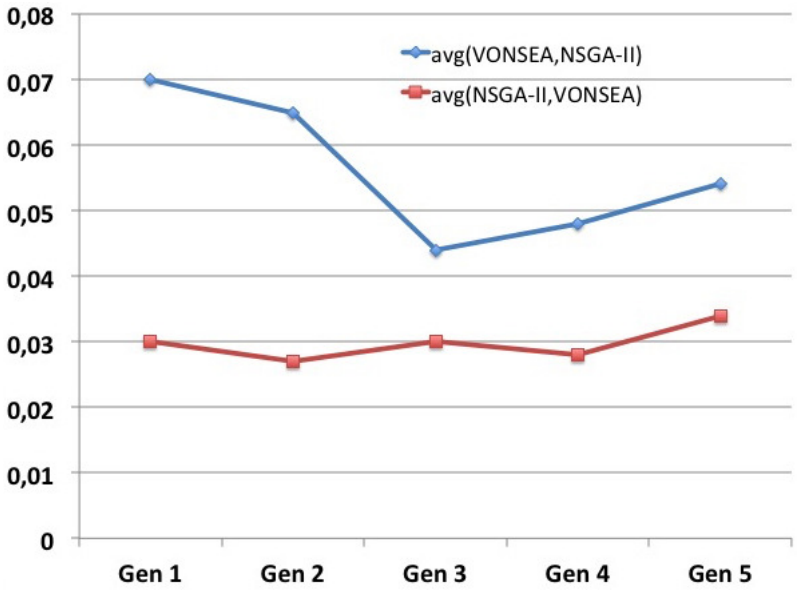

Fig. 6.158: Comparison of the detected non-dominated solutions of VONSEA and NSGA-II per generation by the average number of dominated solutions per solution in each other's set of non-dominated solutions. (4D-MOP)

solutions detected in the test runs by the average number of dominated solutions per solution in each other's set of non-dominated solutions in each generation, using the indicator (6.1). Figure 6.158 presents a performance superiority of the non-dominated solutions of VONSEA in each generation compared to those of NSGA-II, since the indicator values avg(VONSEA, NSGA-II) are significantly higher than avg(NSGA-II, VONSEA) in each generation. Also in case of the 4D-MOP, each non-dominated solution detected by VONSEA dominates on average a considerably higher number of non-dominated solutions detected by NSGA-II. VONSEA provides a lower and very well-selected number of highly qualified candidate peptides, whereas NSGA-II detects a large number of candidate peptides revealing a wide quality range. This confirms the hypothesis that the search performance of VONSEA is more efficient and target-orientated than the one of NSGA-II.

6.5.4 Discussion

The last series of test refers to the comparison of the VONSEA configuration with the advisable components and parameter settings to the state-of-the-art NSGA-II with a diversity-preserving selection strategy in terms of convergence, diversity and progress regarding the non-dominated solution progress in the direction of the optimal solutions. The results matrices present confidence limits for the means of the NSGA-II and VONSEA performance and confidence limits for the mean of the non-dominated solution number detected by NSGA-II and VONSEA in the case of the 3D- and 4D-MOP are given in Table 6.14. The convergence indicator values ACV_{scaled} of NSGA-II as well as VONSEA

3D	NSGA-II	VONSEA
ACVscaled	[0.54, 0.64]	[0.53, 0.61]
Diversity	[2.60, 3.09]	[3.53, 4.26]
ACVrel	[0.10, 0.19]	[0.04, 0.10]
av. no. NDS	[32, 39]	[13, 15]

4D	NSGA-II	VONSEA
ACVscaled	[0.16, 0.20]	[0.16, 0.20]
Diversity	[4.31, 4.87]	[4.86, 5.37]
ACVrel	[0.09, 0.17]	[0.03, 0.08]
av. no. NDS	[37, 42]	[27, 31]

Table 6.14: Results matrix of VONSEA and NSGA-II performances and the number of non-dominated solutions detected in the first five generations in the case of the 3D- and 4D-MOP.
Backgrounds: optimal indicator results (green), similar indicator results (blue), worst performance (blue).

are similar or on the same level in the case of the 3D-MOP and 4D-MOP. Otherwise, VONSEA achieves a significantly increased diversity compared to NSGA-II, though a diversity-preserving method is used to determine the individuals of the succeeding generation. Moreover, the relative non-dominated solution quality of VONSEA is considerably better than the one of NSGA-II in the case of the 3D- and 4D-MOP, as indicated by the confidence limits of the ACV_{rel} values. A further analysis of the non-dominated solutions detected by NSGA-II and VONSEA in the first five generations reveals that VONSEA achieves a lower and well-selected number of candidate peptides with a higher quality on average compared to NSGA-II in each of the first five generations and for both MOPs.

In general, these results confirm the preferred VONSEA property of early convergence in contrast to the state-of-the-art NSGA-II. Furthermore, VONSEA confirms the statement of Singh [145] regarding an observable MOGA progress in each successive generation by providing a specific number of highly qualified candidate peptides.

The evolutionary concept of VONSEA has been applied in the practical re-

search work of OPTOPROBE. This project aims to discover fluorescent and peptide-based probes to detect tumor cells of a very small size ($< 1mm$) by an innovative multi-channel fluorescence endoscopy system [69]. Such a peptide-based probe has to comply with several molecular objectives like an optimal binding to the desired target, a high stability and optimized cell permeability. Algorithms based on the analysis of the peptide sequence are available to predict some of these molecular objects. For the evolution of the probes, the evolutionary process has usually performed for five generations and the non-dominated solutions of each generations have been manually inspected regarding further desired molecular properties as fluorescent probes. Summarizing from these practical experiences with the evolutionary concept of VONSEA and the experimental results presented in this section, the benefit of VONSEA compared to the NSGA-II in the case of biological and pharmaceutical research lies in the potential of VONSEA to:

- provide a well-selected number of non-dominated solutions of a high quality per generation instead of a significantly higher number of candidate peptides detected by NSGA-II revealing a wide - and on average worse - quality range.
- provide an chosen number of highly qualified peptides in the very first generation of the evolutionary process. The optimization progress is considerably high in the first five generations. In contrast, the detected candidate peptides of NSGA-II require a further reasonable selection achievable by additional generation runs.

As a consequence, VONSEA selects a modest number of highly qualified peptides in the first few generations and therefore in a robust search process, which subsequently allows an effective and therefore cost-effective manual inspection in the laboratory.

7 Conclusion

The main contribution of this thesis is the development of a MOEA for peptide optimization. The proposed VONSEA has been designed with the aim of providing a considerable number of high-quality peptides within a low number of generations. The configuration of VONSEA as well as its components have been designed based on theoretical considerations. The different optional VONSEA configurations and the common EA objectives, such as selection pressure and the influence of multi-parent recombination on the algorithm performance, have been investigated empirically on two molecular optimization problems of different dimensions. The hypotheses and questions raised in the Section 'Research Issues' are summarized and discussed in the following:
Hypothesis 1 (H1) concerns the role of mutation and recombination as well as their interdependence on the performance of VONSEA. The experiments allow the interpretation that the algorithm performance of VONSEA in solving the 3D- and 4D-MOP is mainly influenced by the mutation operators. A characteristic convergence and diversity behavior is observable for each mutation operator, and these results are nearly independent of the four presented recombination operators and similar for the 3D- and the 4D-MOP.
The first question (Q1) concerns an advisable combination of the proposed seven mutation and four recombination operators that potentially provide optimized performance for the purpose of peptide optimization investigated by the 3D and 4D-MOP. Two combinations of variation operators are proposed as advisable default settings: the adapted mutation operator of Bäck and Schütz in combination with either the linear or the exponential recombination operator. These combinations provided a good convergence-diversity balance for both MOPs.
Hypothesis 2 (H2) concerns the influence of multi-parent recombination on convergence and diversity performance in the case of the 3D- and 4D-MOP. Moreover, the interdependence between parent number and population size is investigated in the case of two different recombination operators, LiDeRP and ExpoDeRP. The experiment results reveal that the influence of the parent number on the performance is recombination-specific but comparable for both

MOPs. In general, an increase of the parent number higher than 3 mostly results in diversity improvement at the cost of convergence, as a consequence of the high sequence disruption and a simultaneous high difference in appearance of the offspring to their parents. Promising and robust performance results are achieved for a parent number of 3 and a population size $ps \geq 100$ in the case of LiDeRP. Acceptable and robust performance results are achieved for a parent number smaller than 5 and a population size of 100 and higher in the case of ExpoDeRP.

Hypothesis 3 (H3) refers to the guidance of the search process by the selection strategies. The three proposed selection strategies combine the aspects of change by tournament selection and the preference of high-quality solutions by assigning higher selection probabilities to these individuals by random selection from the first front in one case and a more discerning indicator-based selection in the other. Diversity within the solutions is implied in the selection process by fitness-proportionate front-based SUS. The experimental results allow the conclusion that front-based SUS, as compared to the random selection of a solution for the succeeding generation, provides a good search performance regarding the convergence-diversity balance. Generally, the selection strategies aggregate selection and ACV-based selection are comparable for convergence and diversity by a suitable probability balance between front-based SUS selection and individual selection from the first front or selection of the individual with the best indicator value. The difference between aggregate and ACV-based selection is the more discerning selection of ACV-based selection which results in remarkably better relative non-dominated solution quality. The properties specific for the selection strategy are observable in the 3D- as well as 4D-MOP.

The second question (Q2) concerns the selection parameters tournament size (ts) and the probability (p_0) of selecting individuals from the first front or the individual with the best indicator value. The parameter p_0 is mainly responsible for convergence-diversity balance, independent of the ts settings and of the problem dimension. But the parameter setting of p_0 depends on the selection criterion for the selection of high-qualified solutions. An advisable parameter setting for p_0 independent of problem dimension is given. The parameter ts is responsible for a more distinctive selection. Independent of the problem dimension and the parameter setting of p_0, a ts setting of at least 8 or higher usually achieves a good convergence-diversity balance.

Question 3 (Q3) concerns a potential convergence improvement by an increase of the population size (ps). A continuous improvement of the convergence performance by an increase of ps is contrary to the early convergence property.

Convergence as well as relative non-dominated solution quality is improved by an increase of ps for all selection strategies, but the level of improvement decreases remarkably or even stagnates for ps settings above 100. Otherwise, diversity is not improvable by an increase of ps or even reducible with ps settings above 100. Therefore, a convergence improvement by an unlimited increase of ps is not observable. These observations are similar for the 3D- and 4D-MOP.

The fourth question (Q4) concerns a possible interdependence between population size and the selection strategies. The experiment results reveal no general interdependence between the three selection strategies and ps. The influence of different ps settings on performance is similar for the three selection strategies and even for the 3D- and 4D-MOP.

Question 5 (Q5) is a logical consequence of Q3 and Q4 and concerns an advisable range of ps settings which potentially provide robust algorithm performance. Advisable ps settings have been stated in the range of 90 to 110 for the 3D- as well as the 4D- MOP.

These results imply the answer to Question 6 (Q6): Since the influence of the different ps settings on algorithm performance is similar for the three selection strategies and independent of the selection parameter settings, the advisable selection parameters are not influenced by the advisable population sizes. The experimental results reveal that there is a superficial influence of the selection parameter tournament size on algorithm performance. A good performance is generally achieved for at least $ts = 8$ or higher. The higher the ts settings, the more discerning are the selection processes, but computational complexity increases. This tendency is observable for all selection configurations as well as for the 3D- and 4D-MOP. The selection parameter p_0 determines the balance between the probability of selecting high quality solutions and the selection of solutions by front-based SUS. The advisable p_0 settings depend on the selection strategies, but are equal for the 3D- and 4D-MOP, providing a robust convergence-diversity balance.

Question 7 (Q7) refers to the characterization of the 3D- and 4D-multi-objective landscapes and the resulting challenges of a MOEA in solving these problems. The characteristic features have been analyzed with regards to modality, ruggedness and characterization of plateaus. In general, both landscapes are very rugged and reveal no specific or even structure. Furthermore, this landscape analysis reveals the common fact that the number of non-dominated solutions increases with the dimension increase and front diversity is reduced. This observation is challenging for a MOEA, and the design of its selection strategy has to take account of this fact. Moreover, the existence of the considerable

number of front-based plateaus spread over the 3D- and 4D-MOML suggests the common approach of balancing the search behavior of the MOEA towards exploration in early generations and exploitation in later generations. This balance is performed by variation operators in a MOEA.

7.1 Summary of Results

The basic results of this thesis are summarized in the following:

1. The presentation of the multi-objective evolutionary algorithm VONSEA, especially designed for peptide optimization. The configuration of VONSEA as well as the variation operators and the selection strategies have been adapted or invented based on theoretical considerations. These considerations are firstly an exemplary multi-objective molecular landscape analysis on two different dimensional and generic molecular optimization problems. Furthermore, different established components have been discussed as to their respective advantages. Some of these components have been adapted and additional components have been introduced for the purpose of peptide optimization.

2. The introduction of a concept for qualitative real-valued multi-objective molecular landscape analysis. The analysis techniques arising from single-objective landscape analysis have been transferred and re-defined for MOMLs. This concept has been used to analyze the two different dimensional generic molecular optimization problems with the aim of formulating hypotheses regarding the design of a MOEA with optimized search performance.

3. The introduction of a global convergence indicator ACV that allows the statistically reasonable evaluation of populations with different sizes. The properties of this indicator have been discussed and a further indicator based on ACV has been introduced as an indicator for the non-dominated solution quality relative to the entire population.

4. The presentation of the experimental results of the systematically investigated different VONSEA configurations as well as a fine-tuning of the selection parameters. Furthermore, the common EA objectives such as selection pressure and the influence of the parent number variation for

recombination have been systematically analyzed. The experiments reveal a MOEA that provides a potentially robust performance for peptide optimization problems of different sizes.

5. The VONSEA configuration with the advisable components and parameter settings has been compared to the state-of-the-art NSGA-II with a diversity-preserving method for the determination of the individuals for the succeeding generation. In general, VONSEA outperforms NSGA-II in terms of convergence, diversity and relative non-dominated solution quality in the case of the 3D- and 4D-MOP. Furthermore, in the first five generations VONSEA provides a low but specific number of high quality non-dominated solutions, which are on the average of a higher quality than the non-dominated solutions provided by NSGA-II.

7.2 Future Research Directions

VONSEA provides an exemplary robust performance for a 3D- and 4D-MOP. To manifest the efficiency and the robustness of VONSEA, its performance needs to be investigated further on a variety of other molecular optimization problems, especially for high-dimensional optimization problems with more than four objectives. Further experiments will establish VONSEA as an optimized metaheuristic in the field of peptide optimization if the hypotheses stated in this thesis and exemplarily verified on the 3D- and 4D-MOP will be confirmed for a wide range of molecular optimization problems. If necessary, some adaptations will have to be made in these cases.
A further improvement that is of specific interest for a practical application in the field of peptide optimization is the integration of the principle of archiving into VONSEA to provide the opportunity of a continuous self-controlled peptide improvement. This research niche requires investigation concerning a good balance of the relational size of the internal and external population as well as a suitable archive-update technique. Furthermore, an advisable ratio of individuals from the external and internal population for recombination is challenging. Moreover, a selection strategy is required that ensures a discerning and well-proportional selection of individuals from the external an internal population.
The consequences of the landscape analysis of the 3D- and 4D-MOP allow a further hypothesis that results in a more fundamental restructuring of a basic MOEA component: the number of non-dominated solutions will increase with

the optimization problem dimension and the front diversity will consequently be reduced, as observed in the case of the 3D- and 4D-MOP. This fact makes a domination-based MOEA less effective, especially in the case of a front-based selection strategy. The consequence of this consideration is an adaptation of the domination principle or a replacement of the Pareto dominance principle to achieve a more discerning ranking of the solutions in the case of higher dimensional optimization problems. An alternative concept to the Pareto dominance principle does not exist to date, opening up a further research direction.
From a more general point of view, the components selection strategies and variation operators are abstract enough to be used in other MOEAs to solve a variety of other real-world and real-valued optimization problems. Extensive experiments in this direction are desirable and of specific interest for the exploitation of the abstract ideas of these operators for other MOEA applications.

List of Figures

List of Tables

References

[1] Aggarwal, s., Garg, R., and Goswami, P. A Review Paper on Different Encoding Schemes used in Genetic Algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering 4*, 1 (2014), 596–600.

[2] Aguirre, H., and Tanaka, K. Working Principles, Behavior, and Performance of MOEAs on MNK-Landscapes. *European Journal of Operational Research 181*, 3 (2007), 1670–1690.

[3] Albuquerque, P., Chopard, B., and Mazza, C. On the Impact of the Representation on Fitness Landscape. *Proc. of the Third International Conference on Genetic Programming* (2000), 1–15.

[4] Babbar, A., Lakshmikantha, A., and Goldberg, D. A Modified NSGA-II to Solve Noisy Multiobjective Problems. *In Proceedings of the Genetic and Evolutionary Computation Conference, AAAI* (2003), 21–27.

[5] Bäck, T. The Interaction of Mutation Rates, Selection and Self-adaption within a Genetic Algorithm. *Proc. of the 2nd Parallel Problem Solving from Nature* (1992), 85–94.

[6] Bäck, T. Optimal Mutation Rates in Genetic Research. *Proc. of the Fifth International Conference on Genetic Algorithms* (1993), 2–8.

[7] Bäck, T. Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. *First IEEE Eonference on Evolutionary Computing 1* (1994), 57–62.

[8] Bäck, T. *Evolutionary Algorithms in Theory and Praxis.* New York: Oxford Univ. Press, 1996.

[9] Bäck, T. An Overview of Parameter Control Methods by Self-Adaption in Evolutionary Algorithms. *Fandamenta informaticae 34*, 1-15 (1998).

[10] Bäck, T., and Hoffmeister, F. Extended Selection Mechanisms in Genetic Algorithms. *ICGA4* (1991), 92–99.

[11] Bäck, T., and Schütz, M. Intelligent Mutation Rate Control in Canonical Genetic Algorithm. *Proc. of the International Symposium on Methodology for Intelligent Systems* (1996), 158–167.

[12] Bäck, T., and Schwefel, H.-P. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation 1*, 1 (1993), 1–23.

[13] Baker, J. Adaptive Selection Methods for Genetic Algorithms. *In Proceedings of an International Conference on Genetic Algorithms* (1985), 101–111.

[14] Baker, J. Reducing Bias and Inefficiency in the Selection Algorithm. *Proceedings of the Second International Conference on Genetic Algorithms and their Application (Hillsdale, New Jersey: L. Erlbaum Associates)* (1987), 14–21.

[15] Bechikh, S., Belgasim, N., and Ben Said, L. Ghédira, K. PHC-NSGA-II: A Novel Multi-objective Memetic Algorithm for Continuous Optimization. *Proc. of the 20th Int. Conf. on Tools with Artifical Intelligence, IEEE* (2008), 180–189.

[16] Beume, N., Fonseca, C., Lopez-Ibanez, M., Paquete, L., and Vahrenhold, J. On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation 13*, 5 (2009), 1075–1082.

[17] Beume, N., Naujoks, B., and Emmerich, M. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *European Journal of Operation Research 181 (3)* (2007), 1653–1669.

[18] Bhardwaj, A., Leelavathi, S., Mazumdar-Leighton, S., Gosh, A., Ramakumar, S., and Reddy, V. The Critical Role of N- and C-Terminal Contact in Protein Stability and Folding of a Family 10 Xylanase under Extreme Conditions. *PLos ONE 5*, 6 (2010), e11347.

[19] Blickle, T., and Thiele, L. A Comparison of Selection Schemes used in Genetic Algorithms. *TIK-Report, Zürich* (1995).

[20] Bradstreet, L., While, L., and Barone, L. A Fast Incremental

Hypervolume Algorithm. *IEEE Transaction on Evolutionary Computation 12*, 6 (2008), 714–723.

[21] Bridges, C., and Goldberg, D. An Analysis of Reproduction and Crossover in a Binary-Coded Genetic algorithm. *2nd International Conference on Genetic algorithms and their Applications.* (1987), 9–13.

[22] Brinkmann, K. An Improved Algorithms for Klee's Measure Problem on Fat Boxes. *Computational Geometry: Theory and Applications 45* (2012), 225–233.

[23] Brockhoff, D., Wagner, T., and Trautmann, H. On the Properties of the R2 Indicator. *In Genetic and Evolutionary Computation Conference (GECCO 2012)* (2012), 465–472.

[24] Cardoso, R., da Cruz, A., Wanner, E., and Takahashi, R. Multi-objective Evolutionary Optimization of Biological Pest Control with Impulsive Dynamics in Soybean Crops. *Bulletin of Mathematical Biology 71*, 1463- 1481 (2009).

[25] Caruana, R., Eshelman, L., and D., S. Representation and Hidden Bias II: Eliminating Defining Length Bias in Genetic Search via Shuffle Crossover. *In 11-th Int. Joint Conf. on Artificial Intelligence 1* (1989), 750–755.

[26] Carvalho, A., and A.F., A. Improving NSGA-II with an Adaptive Mutation Operator. *Proc. of the 11th Annual Conference Companion on Genetic and Evolutionary Computational Conference, GECCO'09* (2009), 2697–2700.

[27] Castranho, M., and Santos, N. *Peptide Drug Discovery and Development.* Wiley-VCH, 2011.

[28] Ceperly, D., Chen, Y., Crain, R., Meng, X., Mira, A., and Rosenthal, J. Challenges and Advances in High Dimensional and High Complexity Monte Carlo Computation and Theory. Workshop, Banff International Research Station for Mathematical Innovation Discovery, 2012.

[29] Chan, T. A Slightly Faster Algorithm for Klee's Measure Problem. *Computational Geometry: Theory and Applications 43*, 243-250 (2010).

[30] Coello Coello, C., and Cruz Cortis, N. Solving Multiobjecti-

ve Optimization Problems using an Aritifical Immune System. *Genetic Programming Evolvable Machines 6*, 2 (2005), 163–190.

[31] Corne, D., Jerram, N., and Knowles, J. Region-based Selection on Evolutionary Multiobjective Optimization. *Proceedings of the Genetic and Evolutionary Conference GECCO'01* (2001), 283–290.

[32] Corne, D., Knowles, J., and Oates, M. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. *Proceedings of Parallel Problem Solving from Nature PPSN VI 1917* (2000), 839–848.

[33] Cutello, V., Narzisi, G., and Nicosia, G. Computational Studies of Peptide and Protein Structure Prediction Problem via Multiobjective Algorithms. *Multiobjective Problem Solving from Nature, Springer Berlin Heidelberg* (2008), 93–114.

[34] Davis, L. Applying Adaptive Algorithms to Epistatic Domains. *Proc. of the International Joint Conference on Artificial Intelligence.* (1985), 162–164.

[35] De Jong, K., and Spears, W. A Formal Analysis of the Role of Multi-point Crossover in Genetic Algorithm. *Annals of Mathematics and Artificial Intelligence 5* (1992), 1–26.

[36] Deb, D., and Deb, K. Investigation of Mutation Schemes in Real-Parameter Genetic Algorithms. *In Proceedings of Swarm, Evolutionary and Memetic Computing, SEMCCO 2012 7677* (2012), 1–8.

[37] Deb, K. *Multi-objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.

[38] Deb, K., and Agarwal, R. Simulation Binary Crossover for Continous Search Space. *Complex Systems 9(2)* (1995), 115–148.

[39] Deb, K., and Agrawal, A. A Niched-penally Approach for Constant Handling in Genetic Algorithms. *In Proceedings of the International Conference on Artificial Networks and Genetic Algorithms, ICANNGA-99* (1999), 235–243.

[40] Deb, K., and Jain, H. Handling Many-Objective Problems using an Improved NSGA-II Procedure. *Evolutionary Computation, CEC '12* (2012), 1–8.

[41] Deb, K., and Jain, S. Running Performance Metrics for Evolutionary Multiobjective Optimization. *Kan GAL Report No. 2002004, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur* (2002).

[42] Deb, K., Joshi, D., and Anand, A. Real-coded Evolutionary Algorithms with Parent-centric Recombination. *KanGAL Report No. 2001003* (2001).

[43] Deb, K., Mohan, M., and Mishra, S. A Fast Multi-Objective Evolutionary Algorithm for Finding Well-Spread ParetoOptimal Solutions. *KanGAL Report No. 2003002, Indian Institute of Technology, Kanpur, India* (2003).

[44] Deb, K., Pratap, A., and Agarwal, S. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6 (2)* (2002), 182–197.

[45] Deb, K., and Reddy, A. Reliable Classification of Two-Class Cancer Data using Evolutionary Algorithms. *BioSystems 72* (2003), 111–129.

[46] Deb, K., and S., T. Omni-Optimizer: A Genetic Evolutionary Algorithm for Single and Multi-objective Optimization. *European Journal of Operational Research 185* (2008).

[47] Deb, K., Sinha, A., and Kukkonen, S. Multi-objective Test Problems, Linkages and Evolutionary Methologies. *Proceedings of the 8-th Annual Conference on Genetic and Evolutionary Computation* (2006), 1141–1148.

[48] D'Souza, D., Seharan, K., and A., K. Improved NSGA-II based on Novel Ranking Scheme. *Journal of Computing 2*, 2 (2010), 91–95.

[49] Dumitrescu, D., Lazzerini, B., Jain, L., and Dumitrescu, A. Evolutionary Computation. *CRC Press LLC* (2000).

[50] Durillo, J., and Nebro, A. jmetal: A Java Framework for Multi-objective Optimization. *Advances in Engineering Software 42*, 10 (2011), 760–771.

[51] Edwards, C., Cohen, M., and Bloom, S. Peptides as Drugs. *QJM: An International Journal of Medicine 92 (1)* (1999), 1–4.

[52] Eiben, A., and Bäck, T. An Empirical Investigation of Multi-parent Recombination Operators in Evolutionary Strategies. *Evolutionary Computation 5*, 3 (1997), 347–365.

[53] Eiben, A., Raué, P.-E., and Ruttkay, Z. Genetic Algorithms with Multi-Parent Recombination. *In Proceedings of the third Conference on Parallel Problem Solving from Nature - PPSNIII 866* (1994), 78–87.

[54] Emmerich, M., Beume, N., and Naujoks, B. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. *EMO 2005, LNCS 3410* (2005), 62–76.

[55] Emmerich, M., Lee, B., Render, A., Faddiev, E., Kruisselbrink, J., Deutz, A., van der Horst, E., IJzerman, A., and Bäck, T. Analyzing Molecular Landscapes Using Random Walks and Information Theory. *Chemistry Central Journal 3*, 1 (2009), 20.

[56] Engelbrecht, A. *Computational Intelligence: An Introduction.* John Wiley& Sons, 2007.

[57] Erickson, M., Mayer, A., and Horn, J. The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. *Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization EMO 2001 Lecture Notes in Computer Science 1993* (2001), 681–695.

[58] Eshelman, L., Caruana, R., and D., S. Biases in the Crossover Landscape. *Proc. 3rd. Int. Conf. on Genetic Algorithms, ed. J.D. Schaffer 3* (1989), 10–19.

[59] Eshelman, L., Mathias, K., and Schaffer, J. Crossover Operator Biases: Exploiting the Population Distribution. *Proc. ICGA 97* (1997), 354–361.

[60] Eshelman, L., and Schaffer, J. Real-coded Genetic Algorithms and Interval Schemata. *In D. Whitley (Ed.), Foundation of genetic Algorithm II* (1993), 187–202.

[61] Fahrmeier, L., Künstler, R., Pigeot, I., and Tutz, G. *Statistik: Der Weg zur Datenanalyse.* Berlin; Springer, 1999.

[62] Fang, H., Wang, Q., Tu, Y., and Horstemeye, M. An Efficient Non-dominated Sorting Method for Evolutionary Algorithms. *IEEE*

Trans. on Evolutionary Computation 16, 3 (2008), 355–384.

[63] Fogarty, T. Varying the Probability of Mutation in Genetic Algorithm. *Proc. of the Third International Conference on Genetic Algorithms* (1989), 104–109.

[64] Fogel, L., Owens, A., and Walsh, M. *Artificial intelligence through simulated evolution.* John Wiley, 1966.

[65] Fonseca, C., and Fleming, P. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proc. of the Fifth Int. Conf. on Genentic Algorithms, (Forrest Ed.)* (1993), 416–423.

[66] Fonseca, C., and Fleming, P. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms - Part ii: Application Example. *IEEE Trans. Syst. Man, Cybern. A. 28* (1998), 38–47.

[67] Fonseca, C., Paquete, L., and Lopez-Ibanez, M. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. *In 2006 IEEE Congress on Evolutionary Computation (CEC'2006)* (2006), 3973–3979.

[68] Fortin, F.-A., and Parizeau, M. Revisiting the NSGA-II Crowding-Distance Computation. *Proc. of the 15th Annual conference in Genetic and Evolutionary Computation, GECCO '13* (2013), 623–630.

[69] Frey, A. Schlussbericht zum Forschungsvorhaben 'Evolvierung von Peptidliganden und Erforschung intelligenter Sonden für die optische Bildgebung' im Forschungsverbund 'Optischen Sonden für die medizinische Diagnostik und die zellbiologische Forschung': OPTOPROBE. *Technische Informationsbibliothek und Universitätsbibiliothek, Hannover*, DOI: 10.2314/GBV:813174007 (2013).

[70] Garrett, D., and Dasgupta, D. *Multiobjective Landscape Analysis and the Generalized Assignment Problem.* Springer-Verlang, Berlin, Heidelberg, 2008, pp. 110–124.

[71] Garrett, D., and Dasgupta, D. Plateau Connection Structure and Multiobjective Metaheuristic Performance. *Congress on Evolutionary Computation, (CEC 2009)* (2009), 1281–1288.

[72] Ghosh, A., and Dehuri, S. Evolutionary Algorithms for Multi-

Criterion Optimization: A Survey. *International Journal of Computing & Information Science 2*, 1 (2004), 38–57.

[73] Gillet, V., Willet, P., Fleming, P., and D.V.S., G. Designing focused Libraries using MoSELECT. *J. Mol. Graph Model 20* (2002), 491–498.

[74] Goh, K., Lim, A., and Rodrigues, B. Sexual selection for Genetic Algorithms. *Artificial Intelligence Review, Kluwer Academic Publisherr 19*, 123-152 (2003).

[75] Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning. *Addison Verlag* (1989).

[76] Goldberg, D., and Deb, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithm. *In Foundation of Genetic Algorithm* (1991), 69–93.

[77] Goldberg, D., and Lingle, R. Alleles, Loci, and the Traveling Salesman Problem. *Proc. of the First International Conference on Genetic Algorithms and their Applications.* (1985), 154–159.

[78] Guruprasad, K., Reddy, B., and Pandit, M. Correlation between Stability of a Protein and its Dipeptide Composition: A Novel Approach for predicting In Vivo Stability of a Protein from its Primary Structure. *Protein Engineering 4*, 2 (1990), 155–161.

[79] Hajela, P., and Lin, C.-Y. Genetic Search Strategies in Multicriterion Optimal Design. *Struct Optimization 4(2)* (1992), 99–107.

[80] Hansch, C., Björkroth, J., and Leo, A. Hydrophobicity and Central Nervous System Agents: On the Principle of Minimal Hydrophobicity in Drug Design. *J. Pharmacol. Sci. 76*, 9 (1987), 663–687.

[81] Henikoff, S., and Henikoff, J. Amino Acid Substitution Matrices from Protein Blocks. *Proc. Natl. Acad. Sci. USA 89*, 22 (1992), 10915–10919.

[82] Herrera, F., Lozano, M., and Verdegay, J. Tackling Real-coded Genetic Algorithms: Operators and Tools for Behavioral Analysis. *Artificial Intelligence Review, Kluwer Academic Publisherr 12* (1998), 265–319.

[83] Hohm, T., Limbourg, P., and Hoffmann, D. A multi-objective Evolutionary Algorithm for the Design of Peptide Mimotopes. *Journal of Computational Biology 13*, 1 (2006), 113–125.

[84] Holland, J. Adaption in Natural and Artificial Systems. *Ann Arbor: University of Michigan Press* (1975), 287–299.

[85] Hopp, T., and Woods, K. Prediction of Protein Antigenic Determinants from Amino Acids Sequences. *In Proc. Natl. Acad. Sci USA 78* (1981), 3824–3828.

[86] Hopp, T., and Woods, K. A Computer Programm for Predicting Protein Antigenic Determinants. *Mol. Immunology 20*, 4 (1983), 483–489.

[87] Horn, J., Nafpliotis, N., and Goldberg, D. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. *IEEE International Conference on Evolutionary Computation 1* (1994), 82–87.

[88] Hortijk, W. A Measure of Landscapes. *Evolutionary Computation 4*, 4 (1996), 335–360.

[89] Jones, G. *Genetic and Evolutionary Algorithms*. Encyclopedia of Computational Chemistry, John Wiley & Sons, Ltd., 1998.

[90] Jones, T., and Forrest, S. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithm. *In Proceedings of the 6th International Conference on Genetic Algorithms* (1995), 184–192.

[91] Kershenbaum, A. When Genetic Algorithms Work Best. *Journal of Computing, INFORMS 9*, 3 (1997), 254–255.

[92] Khare, V., Yao, X., and Deb, K. Performance Scaling of Multi-objective Evolutionary Algorithms. *Proc. of the Second Int. Conf. on Evolutionary Multi-Criterion Optimization, EMO 2003 2632*, 376-390 (2003).

[93] Kita, H., and Yamamura, M. A Functional Specialization Hypothesis for Designing Genetic Algorithms. *In Proc. of the IEEE Int. Conf. on systems, Man, Cypernetics* (1999), 579–584.

[94] Knapp, B., Gicziv, V., and Ribarics, R. PeptX: Using Genetic Algorithms to Optimize Peptides for MHC Binding. *BMC Bioinformatics*

12:241 (2011), 1–9.

[95] Knowles, J., and Corne, D. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation 8 (2)* (2000), 149–172.

[96] Knowles, J., and Corne, D. Towards Landscape Analysis to Inform the Design of Hybrid Local Search for the Quadratic Assignment Problem. *Soft computing Systems: Design, Management and Applications* (2002), 271–279.

[97] Konak, A., Coit, D., and Smith, A. Multi-objective Optimization using Genetic Algorithms: A Tutorial. *Reliability Engineering and System Safety 91, Elsevier* (2006), 992–1007.

[98] Köppen, M., Vincente-Garcia, R., and Nicholay, B. Fuzzy-Pareto-Dominance and its Application in Evolutionary Multi-objective Optimization. *Proc. of the 3rd Int. Conf. on Evolutionary Multi-Criterion Optimization 3410* (2005), 399–412.

[99] Kuckartz, U., Rädiker, S., Ebert, T., and Schehl, J. *Statistik: Eine verständliche Einführung.* Springer Verlag, 2010.

[100] Kumar, A. Encoding schemes in genetic algorithm. *International Journal of Advanced Research in IT and Engineering, IJARIE 2*, 3 (2013), 1–7.

[101] Laumann, M., Thiele, L., Deb, K., and Zitzler, E. Combining Convergence and Diversity in Evolutionary Multi-Objective Optimization. *Evolutionary Computation 10*, 3 (2002), 263–282.

[102] Laumanns, L., Zitzler, E., and Thiele, L. A Unified Model for Multi-objective Evolutionary Algorithms with Elitism. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2000) 1* (2000), 46–52.

[103] Lee, B. Analysing Molecular Landscapes using Random Walk and Information Theory. Master's thesis, LIACS, University of Leiden, 2009.

[104] Lee, I.-H., Shin, S.-Y., and Zhang, B.-T. DNA Sequence Optimization Using Constrained Multi-Objective Evolutionary Algorithm. *Evolutionary Computation, CEC '03 4* (2003), 2270–2276.

[105] Leier, A., and Banzhaf, W. Exploring the Search Space of Quantum Programs. *In Proceedings of the 2003 Congress on Evolutionary Computation IEEE Press 1* (2003), 170–177.

[106] Li, M., Zheng, J., and Wu, J. Improving NSGA-II Algorithm based on Minimum Spanning Tree. *Proc. of the 7th Int. Conf. SEAL 5361* (2008), 170–179.

[107] Lu, H., and Yen, G. Rank-Density-based Multiobjective Genetic Algorithm and Benchmark Test Function Study. *IEEE Transactions on Evolutionary Computation 7 (4)* (2003), 992–1007.

[108] Lucasius, C., and Kateman, G. Applications of Genetic Algorithms in Chemometrics. *Proc. of the 3rd Int. Conf. on Genetic Algorithms* (1989), 170–176.

[109] Maccari, G., Di Luca, M., Nifosi, R., Cardarelli, F., Signore, G., Boccardi, C., and Bifone, A. Antimicrobial Peptides Design by Evolutionary Multiobjective Optimization. *PLoS Computational Biology 9*, 9 (2013).

[110] Merkuryeva, G., and Bolshakovs, V. Benchmark Fitness Landscape Analysis. *International Journal of Simulation Systems, Science and Technology 12*, 2 (2011), 38–45.

[111] Merz, P. Advanced Fitness Landscape Analysis and the Performance of Memetic Algorithms. *Evolutionary Computation 12*, 3 (2004), 303–325.

[112] Merz, P., and Freisleben, B. A Genetic Local Search Approach to the Quadratic Assignment Problem. *In Proceedings of the 7th International Conference on Genetic Algorithms* (1999), 465–472.

[113] Michalewicz, Z. *Genetic Algorithm + Data structures = Evolutionary Programs*. "nd ed. Springer Verlag, Berlin, 1994.

[114] Murata, T., and Ishibushi, H. MOGA: Multi-objective Genetic Algorithms. *In: Proceedings of the 1995 IEEE International Conference on Evolutionary Computation 1* (1995), 289–294.

[115] Needleman, S., and Wunsch, C. A General Method Application to the Research for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology 48*, 3 (1970), 443–453.

[116] Nicolaou, C., Brown, N., and Pattichis, C. Molecular Optimization using Computational Multi-objective Methods. *Drug Discovery & Development 10(3)* (2007), 316–324.

[117] Ochoa, G., Harvey, I., and Buxton, H. On Recombination and Optimal Mutation Rates. *In Proc. of Genetic and Evolutionary Computation Conference (GECCO 1999)* (1999), 488–495.

[118] Oduguwa, A., Tiwari, A., Fiorentino, S., and Roy, R. Multi-Objective Optimization of the Protein-Ligand Docking Problem in Drug Discovery. *Genetic and Evolutionary Computation Conference, GECCO 2006* (2006), 1793–1800.

[119] Oliver, I., Smith, D., and Holland, J. A Study of Permutation Crossover Operators on the Traveling Salesman Problem. *Proc. of the 2nd International Conference on Genetic Algorithms (ICGA'87)* (1987), 224–230.

[120] Ono, I., and Kobayashi, S. A Real-coded Genetic Algorithm for Functional Optimization using Unimodal Normal Distribution Crossover. *In Proceedings of th 7th International Conference on Genetic Algorithms (ICGA-7)* (1997), 246–253.

[121] Otvos, L., Ed. *Peptide-based Drug Design.* Springer Verlag, Berlin, 2010.

[122] Picek, S., Jakobovic, D., and Golub, M. On the Recombination Operator in the Real-Coded Genetic Algorithm. *In the Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* (2013), 3103–3110.

[123] Prlic, A., Yates, A., and Bliven, S. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics 28*, 20 (2012), 2693–2695.

[124] Purshouse, R., and Fleming, P. The Multiobjective Genetic Algorithm Applied to Benchmark Problems - An Analysis. *Technical Report No. 796, Department of Automatic Control and Systems Engineering, University of Sheffield, UK* (2001).

[125] Rajapakse, M., Schmidt, B., and Brusic, V. Multi-Objective Evolutionary Algorithm for Discovering Peptide Binding Motifs. *Evo Workshops 2006, LNCS 3907, Springer-Verlag Berlin Heidelberg* (2006), 149–158.

[126] Razali, N., and Geraghty, J. Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. *Proceedings of the World Congress on Engineering, WCE 2011 II* (2011).

[127] Rechenberg, I. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann, 1973.

[128] Reeves, C. *Fitness landscapes.* Search Methodologies, E. Burke and G. Kendall, Eds. Springer, 2005, pp. 587–610.

[129] Reidys, C., and Stadler, P. *Combinatorial Landscape.* SIAM, Rev.44, 2002, pp. 3–54.

[130] Röckendorf, N., Borschbach, M., and Frey, A. Molecular Evolution of Peptide Ligands with Custom-Tailored Characteristics for Targeting of Glycostructures. *PLoS Computational Biology 8*, 12 (2012), 1–10.

[131] Rosenthal, S., and Borschbach, M. A Benchmark on the Interaction of Basic Variation Operators in Multi-objective Peptide Design evaluated by a Three Dimensional Diversity Metric and a Minimized Hypervolume. *M.Emmerich et. al. (eds.): EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV* (2013), 139–153.

[132] Rosenthal, S., El-Sourani, N., and Borschbach, M. Introduction of a Mutation Specific Fast Non-dominated Sorting GA Evolved for Biochemical Optimization. *SEAL 2012 LNCS 7673* (2012), 158–167.

[133] Roubos, J., van Straten, G., and van Boxtel, A. An Evolutionary Strategy for Fed-batch Bioreactor Optimization; Concept and Perfromance. *J. Biotechnol. 67*, 2-3 (1999), 173–187.

[134] Sanchez-Faddeev, H., Emmerich, M., and Verbeek, F. Using Multiobjective Optimization and Energy Minimization to Design an Isoform-selective Ligand of the 14-3-3 Protein. *ISoLA 2012 Part II, LNCS 7610* (2012), 12–24.

[135] Sato, H., Aguirre, H., and Tanaka, K. Controlling Dominance Area of Solutions and its Impact on the Performance of MOEA's. *Evolutionary Multi-Criterion Optimization - EMO 2001, Springer 4403*, 154-166 (2001).

[136] Schaffer, J. Multi-objective Optimization with Vector Evaluated Genetic Algorithms. *In Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms* (1985), 93–100.

[137] Schaffer, J., Caruana, R., L.J., E., and Das, R. A Study of Control Parameters Affecting only Performance of Genetic Algorithms. *In Proceedings of the International Conference on Genetic Algorithms* (1989), 51–60.

[138] Schott, J. Fault Tolerance Design Using Simple and Multicriteria Genetic Algorithms Optimization. *Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge* (1995).

[139] Schrijver, A. *On the history of combinatorial optimization (till 1960).* Handbook of Discrete Optimization, 2005, pp. 1–68.

[140] Schütze, O., Esquivel, X., Lara, A., and Coello Coello, C. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation 16*, 4 (2012), 504–522.

[141] Schwefel, H.-P. Collective Phenomena in Evolutionary Systems. *In P. Checkland and I. Kiss (ed.) Problems of constancy and change - the Complementarity of Systems Approaches to Complexity. International Society for General System Research* (1987), 1025–1033.

[142] Scrinivas, N., and Deb, K. Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation, MIT Press Journals 2 (3)* (1994), 221–248.

[143] Scrinivas, N., and Deb, K. Multi-objective Function Optimization Using Non-dominated Sorting Genetic Algorithms. *Evolutionary Computation 2* (1995), 221–248.

[144] Seada, H., and Deb, K. U-nsga-iii: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results. *Proc. of the 8th International Conference of Evolutionary Multi-Criterion Optimization, EMO 2015*, 34-49 (2015).

[145] Singh, J., Ator, M., and Jaeger, E. Applications of Genetic Al-

gorithms to Combinatorial Synthesis: A Comparative Approach to Lead Identification and Lead Optimization. *J. Am. Chem. Soc. 118* (1996), 1669–1676.

[146] Smith, T., and Waterman, M. Identification of Common Molecular Subsequences. *Journal of Molecular Biology 147* (1981), 195–197.

[147] Snedecor, G., and Cochran, W. *Statistical Methods.* Iowa State University Press, Eighth Edition, 1989.

[148] Solaro, R., Chiellini, F., and Battisti, A. Targeted Delivery of Proteins Drugs by Nanocarriers. *Materials 3* (2010), 1928–1980.

[149] Stadler, P. *Fitness Landscape*, vol. 585. In Lecture Notes in Physics, 2002.

[150] Syswerda, G. Uniform Crossover in Genetic Algorithms. *Proc. ICGA 3* (1989), 2–9.

[151] Tan, P., Steinbach, M., and Kumar, V. *Introduction to Data Mining.* Boston: Pearson Addison Wesley, 2006.

[152] Tettamanzi, A., and Tomassini, M. *Soft Computing: Integrating Evolutionary, Neural and Fuzzy Systems.* Springer Sciences& Business Media, 2013.

[153] Thierens, D. Adaptive Mutation Rate Control Schemes in Genetic Algorithm. *In Proceedings of the 2002 IEEE World Congress on Computational Intelligence: Congress on Evolutionary Computation* (2002), 980–985.

[154] Torchilin, V. Intracellular Delivery of Protein and Peptide Therapeutics. *Drug Discovery Today: Technologies 5*, 2-3 (2008), e95–e103.

[155] Tran, K. An Improved Non-dominated Sorting Genetic Algorithm (ANSGA-II) with Adaptable Parameters. *Int. Jour. of Intelligent Systems, Technologies and Applications 7*, 4 (2009), 347–369.

[156] Trautmann, H., Wagner, T., and Brockhoff, D. Focused Multiobjective Search using R2-Indicator-based Selection. *In Learning and Intelligent Optimization, Springer Berlin Heidelberg* (2013), 70–74.

[157] Tsusui, S., Yamamura, M., and T., H. Multi-parent Recombination with Simplex Crossover in Real-coded Genetic Algorithms. *In Procee-*

dings of the Genetic and Evolutionary Computing Conference (GECCO-99) (1999), 657–664.

[158] Vainio, M., and Johnson, M. Generating Conformer Ensembles using a Multiobjective Genetic Algorithm. *J. Chem. Inf. Model 47(6)* (2007), 2462–2474.

[159] Van Veldhuizen, D. Multiobjective Evolutionary Algorithms: Classification, Analyses and new Innovations. *Ph. D. dissertation, Air Force Insitute of Technology, Dayton, Ohio* (1999).

[160] Van Veldhuizen, D., and Lamont, G. On Measuring Multiobjective Evolutionary Algorithm Performance. *In Proc. of the 2000 Congress on Evolutionary Compuation (CEC 2000)* (2000), 204–211.

[161] Vassilev, V., Fogarty, T., and Miller, J. Information Characteristics and the Structure of Landscapes. *Evolutionary Computation 8*, 1 (2000), 31–60.

[162] Veber, D., Johnson, S., and Cheng, H.-Y. e. a. Molecular Properties that Influence the Oral Bioavailability of Drug Candidates. *J. Med. Chem. 45* (2002), 2615–2623.

[163] Veldhuizen, D., and Lamont, G. Multiobjective Evolutionary Algorithm Test. *In Janice Carroll, Hisham Haddad, Dave Oppenheim, Barrett Bryant and Gery B. Lamont, editors, Proceedings of the 1999 ACM Symposium on Applied Computing, San Antonio, Texas* (1999), 351–357.

[164] Ver, Liefooghe, A., Jourdan, L., and Dhaenens, C. On the Structure of Multiobjective combinatorial Search Space: MNK-Landscapes with Correlation Objectives. *European Journal of Operation Research 227*, 2 (2013), 331–342.

[165] Verel, S., Liefooghe, A., and Dhaenens, C. Set-based Multiobjective Fitness Landscape: A Priliminary Study. *In Proc. of the 13th Conference on Genetic and Evolutionary computation Conference (GECCO'11)* (2011), 769–776.

[166] von der Lippe, M. *Deskriptive Statistik*. Oldenburg Verlag, 2006.

[167] Wagner, T., Beume, N., and Naujoks, B. Pareto-, Aggregation-, and Indicator-Based Methods in Many-objective Optimization. *In: Evo-*

lutionary Multi-Criterion Optimization (EMO 2007) (2007), 742–756.

[168] Wagner, T., Trautmann, H., and Brockhoff, D. Reference Articulation by Means of the R2 Indicator. *In Evolutionary Multi-criterion Optimization (EMO 2013) 7811* (2013), 81–95.

[169] Wang, L., Wang, T.-G., and Luo, Y. Improved Non-dominated Sorting Genentic Algorithm (NSGA)-II in Multi-objective Optimization Studies of Wind Turbine Blades. *Applied Mathematics and Mechanics 32*, 6 (2011), 739–748.

[170] Weinberg, E. Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference. *Biological Cybernetics 63* (1990), 325–336.

[171] While, L., Bradstreet, L., and Barone, L. A Fast Way of Calculating Exact Hypervolume. *IEEE Trans. Evolutionary Computation 10* (2006), 29–38.

[172] While, L., Hingston, P., Barone, L., and Huband, S. A Faster Algorithm for Calculating Hypervolume. *IEEE Transactions on Evolutionary Computation 10*, 1 (2006), 29–38.

[173] Wiese, R., and Glen, E. Rna Structure as Permutation: A GA Approach Comparing Different Genetic Sequencing Operators. *Proc. of the 14th International Symposium ISMIS 2871* (2003), 511–520.

[174] Wright, T., Gillet, V., Green, D., and Pickett, S. Optimizing the Size and Configuration of Combinatorial Libraries. *J. Chem. Inf. Comput. Sci. 43* (2003), 381–390.

[175] Wu, J., and Azarm, S. Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set. *Transactions of the ASME, Journal of Mechanical Design 123* (2001), 18–25.

[176] Xie, H., Zhang, M., and Andreae, P. Another Investigation on Tournament Selection: Modeling and Visualization. *GECCO 2007* (2007), 1468–1475.

[177] Yildiz, H., and Suri, S. On Klee's Measure Problem for Grounded Boxes. *In Proc. ACM Symposium on Computational Geometry (SoCG'12)* (2012), 111–120.

[178] Zhang, Q., and Li, H. A Multi-objective Evolutionary Algorithm ba-

sed on Decomposition. *IEEE Transactions on Evolutionary Computation 11* (2007), 712–731.

[179] Zhong, J., Hu, X., Gu, M., and Zhang, J. Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms. *Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automations* (2005).

[180] Zitzler, E. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. *Ph.D. dissertation, Swiss Federal Inst. of Technology (ETH) Zurich* (1999).

[181] Zitzler, E., Deb, K., and Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comp. 8*, 2 (2000), 173–195.

[182] Zitzler, E., and Künzli, S. Indicator-based Selection in Multiobjective Search. *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII* (2004), 832–842.

[183] Zitzler, E., Laumann, M., and Thiele, L. Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. *Proceedings of the EUROGEN 2001 - Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems* (2001).

[184] Zitzler, E., and Thiele, L. Multiobjective Optimization using Evolutionary Algorithms - A Comparative Case Study. *In A. E. Eiben, T. Bäck, M. Schoenauer and H. P. Schwefel (EDS.), Fifth International Conference on Parallel Problem Solving form Nature (PPSN-V)ck, M. Schoenauer and H. P. Schwefel (EDS.), Fifth International Conference on Parallel Problem Solving form Nature (PPSN-V)* (1998), 292–301.

[185] Zitzler, E., and Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation 3*, 4 (1999), 257–271.

[186] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., and da Fonseca, V. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation 7*, 2 (2003), 117–132.